Helsinki University of Technology
Industrial Information Technology Laboratory Publications
Espoo 2006 TKK-INIT-3

THE APPLICATION OF DYNAMIC VIRTUAL PROTOTYPING TO THE DEVELOPMENT OF CONTROL SYSTEMS

Boris Krassi

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY
TECHNISCHE UNIVERSITÄT HELSINKI
UNIVERSITE DE TECHNOLOGIE D'HELSINKI

Helsinki University of Technology
Industrial Information Technology Laboratory Publications
Espoo 2006 TKK-INIT-3

THE APPLICATION OF DYNAMIC VIRTUAL PROTOTYPING TO THE DEVELOPMENT OF CONTROL SYSTEMS

Boris Krassi

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Department of Computer Science and Engineering for public examination and debate in Auditorium T2 at Helsinki University of Technology (Espoo, Finland) on the 14th of June, 2006, at 12 noon.

Helsinki University of Technology
Department of Computer Science and Engineering
Industrial Information Technology Laboratory

Distribution:
Helsinki University of Technology
Industrial Information Technology Laboratory
P.O. Box 5400 TKK
FIN-02015 Espoo
Finland
Fax: +358-9-451-5351
www.init.hut.fi

ISBN: 951-22-8167-8 (printed version)
ISBN: 951-22-8168-6 (electronic version)
ISSN: 1459-6458

Otamedia Oy
Espoo 2006

HELSINKI UNIVERSITY OF TECHNOLOGY
P.O. Box 1000, FIN-02015 TKK
http://www.tkk.fi

ABSTRACT OF DOCTORAL DISSERTATION

Author: Boris Krassi

Name of the dissertation:

The Application of Dynamic Virtual Prototyping to the Development of Control Systems

Date of the manuscript: January 18, 2006 | Date of the dissertation: June 14, 2006

Monograph

Department: Computer Science and Engineering

Laboratory: Industrial Information Technology

Field of research: computer science, control engineering

Opponent: Professor Juha Röning, Ph.D.

Supervisor: Professor Juha Tuominen, Ph.D.

Abstract:

A new method of developing control systems on the basis of dynamic virtual prototyping (DVP) has been proposed. The method facilitates the control system development process by means of (1) automating the transition from the conventional DVP of a plant to the dynamic model suitable for the control system design and (2) integrating the development process into the overall lifecycle.

The method comprises the three principal stages: (1) representing the plant by its DVP; (2) simulating the DVP and generating the data-based model of the plant; (3) designing the control system using the generated data-based model. Stages 1 and 2 are supported by DVP systems (e.g. IGRIP, LMS/VirtualLab, MSC SimOffice), stage 3 is accomplished by CACSD.

The proposed development method has been adapted to the class of plants that are linearizable, quasi-stationary, stable or stabilizable without using the analytical model and have lumped parameters. The specifics of applying the conventional methods of identification and control system design in the context of DVP have been revealed and analyzed.

The engineering applicability of the method has been proved by means of developing the control system for fine positioning of a gantry crane load.

Keywords: dynamic virtual prototyping, CALS, PLM, CAD, CAE, digital mockup, DMU, IGRIP, control systems, development, design, integration, CACSD, mechatronic systems, gantry crane

Number of pages: 183	
ISBN 951-22-8167-8 (print)	ISSN: 1459-6458
ISBN 951-22-8168-6 (electronic)	

Publisher: Otamedia Oy

Print distribution: Helsinki University of Technology, Industrial Information Technology Lab

The dissertation can be read at http://lib.tkk.fi/Diss/2006/isbn9512281686/

CONTENTS

ACKNOWLEDGEMENT

This dissertation has been written during my time as a researcher with the Industrial Information Technology Laboratory at Helsinki University of Technology. I would like to express my deep gratitude to Professor Juha Tuominen, the head of the laboratory, my teacher and dissertation supervisor. It was Professor Tuominen who had introduced me to the field of virtual prototyping. I sincerely appreciate his constant support, attention and interest to my work.

I wish to thank Professor Viktor Melekhin, Professor Leonid Babko and Professor Roman Stroganov of St. Petersburg State Polytechnical University for reading the manuscript and providing me with numerous useful suggestions that helped to improve the content and clarity of the dissertation. Many thanks to Dr. Kai Zenger of Helsinki University of Technology for his thorough review of the text and constructive feedback.

I am grateful to my pre-examiners Professor Hannu Toivonen of Åbo Akademi and Professor Gunnar Bolmsjö of Lund University for their comments on the methodology, the applied mathematical methods and the interpretation of the results. I would like to express my thankfulness to the opponent, Professor Juha Röning of the University of Oulu, for his time and expertise.

I am obliged to Mr. Heikki Aalto of Delfoi Oy for the assistance with the simulation software IGRIP and to Mr. Hannu Oja of KCI Konecranes Oyj for his kind permission to refer to some technical data.

This work has been partially funded by Helsinki Graduate School in Computer Science and Engineering (special thanks to Professor Martti Mäntylä), the Academy of Finland, Helsinki University of Technology and the Industrial Information Technology Laboratory. All financial support is cordially acknowledged.

I am thankful to my colleagues and friends at the Industrial Information Technology Laboratory and everyone who has helped and supported me in the course of the dissertation work.

Finally, I am truly grateful to my family for their love and encouragement.

Espoo, May 2006

Boris Krassi

LIST OF ABBREVIATIONS

AIC	Akaike's information criterion
BIBO	Bounded-input bounded-output
CAD	Computer-aided design
CAE	Computer-aided engineering
CALS	Continuous acquisition and lifecycle support
CAM	Computer-aided manufacturing
DMU	Digital mockup
DOF	Degree of freedom
DVP	Dynamic virtual prototype, dynamic virtual prototyping
GC	Gantry crane
IGRIP	Interactive graphic robot instruction program (Delmia, Inc.)
ISO	International organization for standardization
LQ	Linear-quadratic optimal control
LQG	Linear-quadratic Gaussian
MDL	Rissanen's minimum description length criterion
MIMO	Multi-input multi-output
PDM	Product data management
PID	Proportional-integral-derivative control
PLM	Product lifecycle management
SISO	Single-input single-output
STEP	Standard for exchange of product data
VP	Virtual prototype, virtual prototyping

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. INTRODUCTION

1.1. Motivation

The modern industry tendencies are characterized by the increasing complexity of technical systems, growing level of customization, decreasing duration of lifecycle, shortening time-to-market, and improving the efficiency and quality of products [Har2001, MA1998, Eff1989].

Information technology plays an important role in responding to these tendencies. The *continuous acquisition and lifecycle support* (CALS) and the *product lifecycle management* (PLM) systems improve and facilitate the technical system lifecycle by providing a uniform information infrastructure to integrate all stages of the technical system lifecycle including design, prototyping, manufacturing, operation, maintenance and recycling. The concept of the *virtual prototype* (VP) or the digital mock-up is the foundation of the integrated lifecycle.

The virtual prototype (digital mock-up, virtual model) is a meta-model, a systemic aggregation of various models describing the geometry of the technical system, its kinematics and dynamics, the aspects of its operation, manufacturing, maintenance and recycling [BKT2004, Das2005]. The *dynamic virtual prototypes* (DVP), which are employed in this dissertation, are a subset of the VPs with the emphasis on the dynamic properties of the modeled system.

CALS (PLM) and the DVP systems are drawing an increasing attention of the scientific, engineering, management and business communities. The most advanced and widely used DVP systems include IGRIP/ENVISION D5 and V5 (Dassault Systèmes, France) [Del2005, Das2005], LMS/VirtualLab (LMS, Belgium) [LMS2005, LMS2001], MSC SimOffice (MSC Software, USA) [MSC2005], Unigraphics NX (Unigraphics, USA) [UGS2005], and ANSYS (ANSYS, USA) [ANS2005].

Today the CALS (PLM) and DVP applications are mostly limited to the conceptual and geometric design, final simulation of the entire technical system, and modeling manufacturing processes. This dissertation is focused on extending the applications of DVP to the development stage including the development of control systems, which play a central role in enhancing the dynamic performance and efficiency of technical systems.

The motivation for employing DVP for the control system development is twofold.

One the one hand, the control engineer confronts the fact that the DVP is gradually becoming the primary source of information about the technical system. This compels the engineer to develop control systems on the DVP basis and to extract the necessary dynamic models directly from the conventional DVP.

On the other hand, the automation dimension of the control system development is addressed by the *computer-aided control system design* (CACSD) systems [Gru2001], whereas the information integration dimension is less supported by the computer-aided tools. However, as it has been shown in this dissertation, it is the information integration of the control system development process into the overall technical system lifecycle that is capable of raising the efficiency of the development process above the level achievable solely by means of the conventional CACSD.

The control system development process can be integrated with other stages of the technical system lifecycle when the conventional DVP is employed throughout the lifecycle and directly utilized as the information basis for obtaining the dynamic model of the plant required for the control system design.

This dissertation has demonstrated that it is difficult to "***extract***" the dynamic model, which is suitable for the control system design, from the conventional structural-parametric DVP. The essence of the difficulties, the so-called "***information barrier***", is that the DVP and the dynamic model have different goals and represent the dynamics information in different ways. As a result, it is necessary to develop the method which would eliminate the information barrier and adapt the control system development to the DVP/CALS lifecycle.

In comparison with the related research, this dissertation provides

1. A ***systematic method*** employing the ***conventional*** DVPs for the ***development*** of control systems and ***integration*** of the development process into the overall lifecycle;
2. An ***implementation and adaptation*** of the method for a ***class*** of plants.

1.2. Objectives and tasks

The ***objective*** of this dissertation is to create a systematic method, which would enable to develop control systems on the information basis of DVP improving the efficiency of the control system development process. The improvement is achieved by

- Automating the transition from the plant DVP to the dynamic model suitable for the control system design;
- Integrating the development process into the overall lifecycle.

The main ***research tasks*** include:

1. Analyze the potential of DVP for modeling dynamic systems. Study the interrelations between DVP and the control system development;
2. Develop and analyze the structure and the main stages of the DVP-based control system development method;
3. Investigate the implications and difficulties of employing the DVP as a basis for the control system development; examine the specifics, limitations and rules of applying the conventional mathematical methods of identification and control system design in conjunction with DVP;
4. Implement and adapt the proposed development method to a class of plants, for which the formal mathematical methods of identification and control system design are available and capable of tackling the DVP-generated data-based model;
5. Demonstrate the engineering applicability of the proposed development method using a representative example; evaluate the proposed method by comparing it with the conventional methods.

1.3. Hypothesis

The hypothesis to be proved in this dissertation is that it is possible and reasonable to develop control systems on the basis of DVP. Specifically, the claims to be proved include:

1. The plant DVP is capable of containing the necessary information on the plant dynamics;
2. The DVP-based development method can be implemented and adapted for a sufficiently wide class of plants using the conventional identification and control system design methods;
3. The employment of DVP improves and facilitates the development of control systems.

1.4. The theoretical and practical contribution of the dissertation

The primary ***theoretical contribution*** includes

1. A new systematic method of developing control systems on the basis of DVP has been proposed and analyzed. The proposed method facilitates the control development process by means of (1) automating the transition from the conventional plant DVP to the dynamic model suitable for the control system design and (2) integrating the development process into

the overall lifecycle. The proposed method requires neither the explicit analytical model nor the physical experiments with the real plant. Also, the method extends the applicability of the contemporary DVP/CALS systems to the development of control systems;

2. The DVP control system development method has been adapted to a selected class of plants. The class includes the plants which are linearizable, quasi-stationary, and stable or stabilizable without using the analytical model and have lumped parameters. The possibility of developing the desired control system with the DVP-based method for a plant which belongs to the selected class has been proved;

3. The specifics, limitations and rules of applying the conventional mathematical methods of identification and control system design in the context of DVP have been revealed and analyzed. The specifics arise from the fact that the DVP and the generated data-based model are employed instead of the analytical model or the real data. The insufficiency of the information on the model uncertainty set and the lack of the formal MIMO control system design methods, which would not require the explicit knowledge of the uncertainty set structure, has been indicated as a bottleneck of the proposed DVP-based method. A procedure for constructing the general form of the uncertain plant on the basis of multiple simulations in the DVP system has been suggested.

4. The engineering applicability of the proposed development method has been proved by means of developing the control system for fine positioning of a gantry crane load and comparing the method results with the results of the conventional methods.

The ***practical contribution*** is that the proposed method

1. Improves the efficiency of the control system development process in terms of increasing its flexibility and reducing the consumption of the time, financial, and intellectual resources;
2. Integrates the development process into the overall technical system lifecycle;
3. Extends the applicability of the DVP systems to the development of control systems.

The proposed DVP-based control system development method can be recommended to the users of the (D)VP/CALS systems, the control engineers, and the developers of the (D)VP/CALS systems. Moreover, the results obtained in this dissertation may be found useful in the education process for the contemporary CAD, CACSD, and (D)VP/CALS technologies.

Some of the main results of this dissertation have been published in [KT2002], [KT2003], [KB2003] and [BKT2004].

1.5. Outline of the dissertation

This dissertation contains seven chapters and two appendices. The structure of this dissertation is as follows.

Chapter 1 is the introduction.

Chapter 2 considers the interplay between DVP and the control system development. It shows the importance of adapting the control system development process to the DVP framework. However, an information “barrier” between the DVP and the development process makes it difficult to “extract” the dynamic model, which would be suitable for the control system design, from the conventional structural-parametric DVP. Therefore, the need for a method (“interface”), which eliminates the barrier and adapts the control system development to the DVP/CALS-integrated lifecycle, is justified.

Chapter 3 elaborates the theoretical and methodological foundation of the proposed development method. The main concept of the method, the description of its principal stages, the analysis of the efficiency and the practical applicability of the method and the adequacy of the developed control system are presented.

Chapter 4 is dedicated to adapting the proposed DVP-based control system development method to a local class of plants. The specifics, limitations and rules of applying the existing mathematical methods of identification and control system design in the context of DVP are analyzed. The following two major groups of methods are considered: the methods, which involve the intermediate identification of the analytical model, and the data-based methods, which do not require such identification. The ways of improving the quality and adequacy of the results are also discussed.

Chapter 5 illustrates the proposed DVP-based method and proves its engineering applicability by means of developing the control system for fine positioning of a gantry crane load.

Chapter 6 presents the general conclusions.

Chapter 7 expands on the future research directions in the domain of DVP and the development of control systems.

The appendices contain some auxiliary calculations and the IGRIP and MATLAB programs referred to in Chapter 5.

CHAPTER 2. MERGING DYNAMIC VIRTUAL PROTOTYPING AND THE DEVELOPMENT OF CONTROL SYSTEMS

In this chapter it will be considered the interplay between *dynamic virtual prototyping* (DVP) and the control system development. The recent advances of DVP and the growing acceptance of the DVP systems in industry compel control engineers to develop control systems on the DVP basis. On the one hand, the DVP becomes the primary representation means of a technical system. The control engineer has to extract the necessary dynamic models from the DVP of a technical system. On the other hand, DVP turns out to be the key to improving the control development process. Hence, the problem of adapting the control system development to the DVP framework proves to be significant and highly relevant.

2.1. Dynamic virtual prototyping

The challenges of the globalization and the sustainable development imply the modern industry tendencies, which can be characterized by the increasing complexity of technical systems, growing level of customization, decreasing duration of the lifecycle, shortening time-to-market, and improving the efficiency and quality of products [Har2001, MA1998, Eff1989].

Information technology plays an important role in responding to these tendencies. The *continuous acquisition and lifecycle support* (CALS) and the *product lifecycle management* (PLM) systems improve and facilitate the technical system lifecycle by providing a uniform information infrastructure to integrate all stages of the technical system lifecycle including design, prototyping, manufacturing, operation, maintenance and recycling. The concept of a *virtual prototype* (VP) or a digital mock-up is the foundation of the integrated lifecycle.

First of all, we have to define the meaning of “virtual”. According to [Vir1996], one of the meanings of “virtual” is “being in such power, force, or effect, though not actually or expressly such”. “Virtual” distinguishes something conceptual from something that has physical reality [Vir2005], or, in other words, something material (i.e. exists in space and time) from something ideal (i.e. exists in only in time, though, requiring a physical medium), for example, a brain vs. mind, a physical embodiment vs. a technical drawing.

In this dissertation, “virtual” is only partially related to the term “virtual reality”, which is defined as “a new paradigm in computer-human interaction, in which three-dimensional computer-generated worlds … have the effect of containing objects that have their own location in three-dimensional space” with the user perception of the virtual world being "…similar to the real world" [Vir2004]. A more precise definition of virtual reality is given in [Vir1994]: "a simulation of an environment that is experienced by a human operator provided with a combination of visual (computer-graphic), auditory, and tactile presentations generated by a computer program".

We define the virtual prototype (digital mock-up, virtual model) as a meta-model, a systemic aggregation of various models describing the geometry of the technical system, its kinematics and dynamics, the aspects of its operation, manufacturing, maintenance and recycling [BKT2004, Das2005]. The VP also takes into account such higher-level phenomena as processes, logistics and so on. On the whole, the VP accumulates all relevant information about the system throughout its lifecycle making the VP concept one of the most advanced approaches to the systemic modeling and simulation of complex technical systems.

The VP is a computer-based substitute of a real system. It is a *system* by itself because the VP has all properties of a system such as integrity (emergence, synergy), purposefulness (a goal or goals), and structuredness (hierarchy). The breadth and depth of the VP allow to consider the VP as a special class of system models[1].

The VP is dual. On the one hand, the VP is a phenomenological model (a grey or a black box) since it appears to the user (engineer) as an integral object rather than a set of mathematical formulae[2]. On the other hand, it is a deductive model because the essence of the VP is an analytical model of some kind (see Section 3.2.1).

In the context of this dissertation, the visual appearance is not the most important aspect of the VP. The distinguishing feature of the VP is that it provides the means for the information integration of the technical system lifecycle. This integration results in the reduced development, manufacturing, operations, and maintenance costs [Bae2002, Rob2002a, Wen2002, Inf2003, Das2005, ZY2003, BDS2003].

The dynamic virtual prototypes (DVP), which are employed in this dissertation, are a subset of the VPs with the emphasis on the dynamic properties of the modeled systems.

Due to the convergence of technologies, there are a number of definitions of the VP concept and neither of those definitions has yet been adopted universally. Let us illustrate the interrelations between CALS (PLM), CAD/CAE/CAM/CACSD and (D)VP, Figure 2.1.

---

[1] In [NY2001] virtual models are classified as the third generation of models: (1) phenomenological, (2) deductive, and (3) virtual. [NY2001] emphasizes the computer-graphic three-dimensional representation of reality. However, in this dissertation the visualization aspect is not assumed to be the most important in the definition of the virtual model

[2] The definition [Vir1994] clearly indicates the phenomenological appearance of the VP

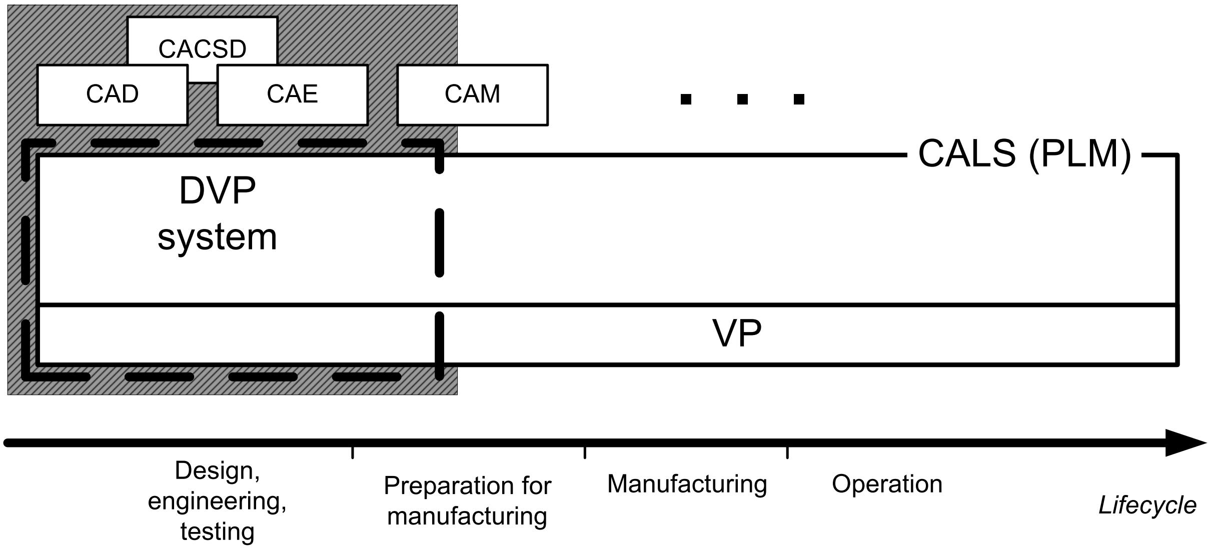

Figure 2.1. Interrelations between CALS (PLM), CAD/CAE/CAM/CACSD and (D)VP. The domain of this dissertation is shown in grey

The CALS (PLM) systems provide the information integration of the technical system lifecycle. The VP is the foundation of CALS (PLM) because, conceptually, it is capable of accumulating all relevant information about the technical system throughout its lifecycle. There are a number of CAD systems, which support the corresponding stages of the lifecycle (design, engineering, manufacturing etc.), residing on the top of the CALS (PLM) layer. By analogy with the distributed computing systems, CALS (PLM) can be treated as a sort of middleware [Bri2001, CDK2001]. The purpose of this middleware is twofold:

- Connecting or "gluing" various CAD systems that support the lifecycle stages (CAD, CAE, CACSD, CAM etc.);
- Isolating or abstracting these CAD systems from the specific form and type of the underlying VP.

A *dynamic virtual prototyping system* (DVP system) is a subset of CALS (PLM), a cross-section of the middleware architecture, which is primarily intended for modeling the processes at the development stage of the lifecycle. One of the very useful functions of the DVP systems is the automated construction of the equations that describe the dynamics of the system based on the information contained in the VP.

In fact, the above explanation is quite rough because the borders between the VP, CALS (PLM), and CAD systems are not well-defined. For example, the PLM systems may comprise some CAD or CAM capabilities, the VP layer can be either included into the CALS (PLM) system or considered as a separate layer and so on. Nevertheless, Figure 2.1 presents a fairly comprehensive picture of the domain.

The most advanced and widely used examples of the DVP systems[3] are IGRIP/ENVISION D5 and V5 [Das2005, Del2005], LMS/VirtualLab [LMS2005], MSC SimOffice (MSC.ADAMS, MSC.Nastran) [MSC2005], Unigraphics NX (former I-deas) [UGS2005] and ANSYS [ANS2005].

Unlike such CAD systems as AutoCAD [Aut2005], 3D MAX [MAX2005] and ProENGINEER [PRO2005], the DVP systems are capable of modeling not only the geometry, but also the kinematics and dynamics of the technical system and its environment. 3D MAX is famous for its advanced 3D animations. In contrast to the DVP systems, which solve the direct kinematics and dynamics problems, 3D MAX only imitates dynamics since motions are preprogrammed. However, the visual appearance of the 3D animations generated by 3D MAX and similar CAD systems is currently far superior to the DVP systems. The 3D dynamic games and simulators resemble the DVP systems because they also solve the direct kinematics and dynamics problems (the motions are not predefined) [EHDS2005]. Yet, the games and simulators are intended for neither design nor the information integration.

Such systems as MODELICA [MCA2005] and the bond graphs [Bon2005] are focused on modeling dynamics. Conversely, the DVP systems are capable of modeling many other aspects (manufacturing etc.) with the emphasis on the phenomenological representation of the modeled system, where the mathematical details are intentionally "hidden" from the user. Therefore, MODELICA and the bond graphs are not true DVP systems. They can be treated as possible dynamic engines for the DVP systems.

Nowadays CALS (PLM) and the DVP systems have been drawing an increasing attention of the scientific, engineering, management and business communities. The DVP systems have been being developed in Europe and the USA by Dassault Systèmes and Delmia (France/USA), MSC Software (USA), LMS (Belgium) and Unigraphics (USA) since the 1970's. Moreover, there are a number of international and national organizations facilitating the CALS/PLM/DVP development, for example, the International Organization for Standardization (ISO) [ISO2005].

The evolution of CALS/PLM in Russia has had a fascinating history, which is virtually unknown in the West. The idea of the comprehensive information support of the entire technical system lifecycle was motivated by the development "Buran-Energia", the Russian space shuttle[4], in the 1970-1980's [LBA1998a, LBA1998a, LBA1999]. Constructing

---

[3] The DVP systems will be discussed at length in Section 4.3.1

[4] The development of CALS/PLM in the USA was also motivated by the space applications, e.g. MSC.Nastran (introduced in the 1960's) stands for "NASA structural analysis program"

"Buran-Energia" was an extremely complex problem. About a thousand construction bureaus, research institutes and factories were involved. The essence of CALS (PLM), or the "virtual enterprise" is very clearly explained in [LBA1998b]: "it was required to create such a structure for the enterprises interaction, which would insure the uniformity of the initial scientific and engineering information and the synchronized timing. The amount of the digitized (vs. paper-based) information had to be maximized". The "virtual enterprise" is defined as "an integrated information structure" based on the uniform mathematical, hardware-in-the-loop, and physical full-scale modeling and simulation [*ibid*].

Today the CALS (PLM) and DVP applications are mostly limited to conceptual and geometric design, final simulation of the entire system, and modeling the manufacturing processes. But the true potential of CALS (PLM) and DVP is in the comprehensive coverage of the entire lifecycle from the concept stage to the recycling. This dissertation is focused on extending the applications of DVP to the development stage including the control system development.

2.2. Structure of the control system development process

Control systems play a central role in enhancing the dynamic performance and efficiency of technical systems. Apart from such traditional applications as aircrafts, nuclear reactors, high precision weapons, today control systems can be found in a multitude of systems from the antilock brake systems (ABS), the active vibration control systems for protecting tall buildings from wind gusts and earthquakes to consumer electronics.

Control can be defined as "the purposeful organization of the processes in nature, technology and society" [PKE1974, p. 5]. If control is implemented without any explicit human intervention, it is called the automatic control [*ibid*]. The ***mathematical theory*** of automatic control systems has been developed since the second half of the 19^{th} century. Automatic control is a principal constituent of cybernetics.

The term "*cybernetics*" was coined in the 1940's [Wie1948]. It is described as "the science of control and communication in all of their manifestations within and between machines, animals, and organizations" [Cyb1994]. A broader definition is given in [Cou1956]: "the art of insuring efficiency of action"[5].

The automatic control theory adopts the information approach rather than the energy one. It exploits the concepts of feedback, dynamic system, optimization, modeling,

[5] "La cybernétique est l'art d'assurer l'efficacité de l'action" [Cou1956, p.52]. Perhaps, the modern definition could be: "the art ***and science*** of insuring the efficiency of action"

identification etc. The plant (object of control) is considered as an integral system described by its functional operator, which maps the input of the plant on its output. *The general control problem*, which resembles one of the inverse dynamics, can formulated as follows: given a plant, find a controller that makes the plant output follow the desired reference input (functional requirements) subject to a number of constraints, for example, limited control authority, precision, and robustness (non-functional requirements).

The development of control systems is an iterative process comprising several phases.

- Studying the plant; formulating the control problem; constructing the dynamic model, which defines the plant dynamics and information properties and abstracts the formal control system synthesis from the specifics of the plant domain; analyzing the dynamic model (pre-analysis);
- Designing[6] the control system (synthesis);
- Analyzing the control system (post-analysis);
- Implementing the control system.

The development process can be represented as a graph with the nodes arranged at four levels[7], Figure 2.2:

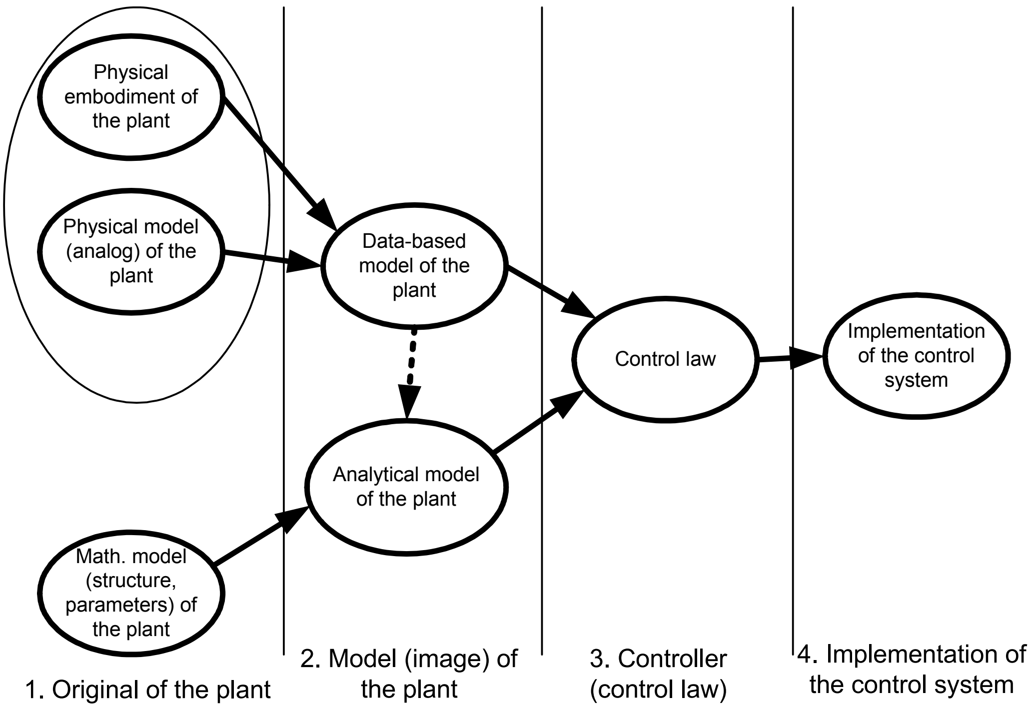

Figure 2.2. Structure of the control system development process

---

[6] Note that ***design*** of the control system is just one of the phases of the control system ***development***

[7] Figure 2.2 shows only the most typical transitions between the graph nodes

- Original of the plant;
- Model (image) of the plant;
- Controller (control law);
- Implementation of the control system.

The model of the plant is a homomorphic *image* of the *original* (the physical plant, its analog or a mathematical model). The image contains only those properties of the plant which are relevant to the control system development. The term *homomorphism* means “similarity of form”, an aggregation of the plant-original elements, or its adequate reduction; this is one of the main stimuli of modeling [Mod1974, Gai2001, Pod2002].

The nodes of the first level fall into two classes depending on the description of the plant-original. First, the original is the plant itself or its physical analog. Second, the original is the mathematical model, i.e. a formalized description of the plant structure and parameters sufficient for constructing the analytical dynamic model.

The nodes of the second level correspond to the analytical, data-based (table input-output form), algorithmic and other types of models.

There are several types of transitions (mappings) between the first (originals) and the second (images) levels. The experiment with the real plant or its physical analog (i.e. physical modeling) leads to a data-based dynamics model. The analytical modeling (i.e. mathematical modeling [SM2001]) results in an analytical (formulae) dynamics model.

From Figure 2.2 it follows that the basis of any method of the control system design is a dynamics model of some kind. Therefore, the control system design methods can be classified in accordance with the underlying models. Traditionally, the following two groups of methods are considered. The first one is based on the analytical model[8], while the second one directly exploits the data-based model.

Nowadays the abstract methods for the analysis and synthesis of control systems are reasonably well developed. The physical implementation of the designed control system is facilitated with a multitude of highly developed software and hardware [Ste2002, Gru2001, BCD2001].

However, it is constructing the model of dynamics that is associated with fundamental difficulties. This phase is hard to formalize since it is strongly coupled with the real physical plant. On the one hand, the analytical modeling requires the highly qualified engineer capable of describing the physical processes of the plant in terms of mathematical formulae [SM2001]. On the other hand, the “physical” methods assume that there is a sufficient (large)

[8] An analytical model can be identified from a data-based model (dotted line in Figure 2.2)

amount of data, which adequately represent the plant in all work regimes of interest. Obtaining those data may be infeasible because the real plant or its analog can be unavailable or expensive to experiment with.

As a result, the efficiency of the control system development process to a certain extent depends on the phase of obtaining the plant dynamics model.

Furthermore, the control system development is just one of the stages of the technical system lifecycle. There are bidirectional information flows between this stage and the stages of the plant design, testing of the whole system, manufacturing, and operations. The control system design is "connected" with other lifecycle stages by means of the plant dynamics model. Hence, if the plant is modified during one of the lifecycle stages, the dynamics model has to be updated. Alternatively, the control engineer may provide some recommendations on improving the plant.

The conclusion is that the control system development stage has to be considered in the context and in conjunction with other stages of the lifecycle.

2.3. Modeling in the context of the control system development

Since modeling plays a crucial role in the control system development, let us consider the mathematical foundations of modeling, similarity and adequacy of models and control systems. This material will be referred to in the subsequent chapters.

In [Mod1994] a *model* is defined as "a mathematical or physical system that is based on knowledge of the structure and function of the object for which the system is designed". Specifically, a mathematical model is "an approximate representation, in mathematical terms, of a concept, an object, a system, or a process" [JK1980].

Modeling is a vast scientific discipline [SM2001]. This section will focus on the aspects of modeling which would enable to find out the practical relationship between the physical plant, its dynamic model and control system, and to assess the adequacy of the designed control systems.

A technical system can be ***represented*** as a formal algebraic system[9] [MT1975, Gai2001]: $S = \langle W, M, R, P \rangle$, where W is the set of the environment elements and their aggregations, M is the set of the system elements and their aggregations, R is the set of the

[9] Although the formalism of algebraic systems is not presented here in a fully rigorous way, this material is useful for illustrating the concepts of model similarity, complexity and adequacy

relationships between the elements, P is the set of the properties of the system and its environment. The relationship between W, M, R, P is as follows:

$$r_i(a,b) = p_j,\ \ a,b \in W \cup M,\ r_i \in R,\ p_j \in P.$$

The behavior of the system can be described by the input set of signals *X*, the output set of signals *Y*, and the functional operator $F : y = F_s(x), x \in X, y \in Y$, which maps the input on the output. If the system is dynamic, i.e. it evolves in time, the functional operator can be written as: $y(t) = F_s(x(t),t)$. Also, any definition of system should explicitly include some constructive goals *Z* [Ano1998, Kol1994]. Finally, for a stationary dynamic system we have:

$$S = \langle W, M, R, P, x(t) \in X, y(t) \in Y, F_s : y(t) = F_s(x(t)), Z \rangle$$

According to [MT1975, Mod1974], system $S_2 = \langle W_2, M_2, R_2, P_2 \rangle$ is a model of system $S_1 = \langle W_1, M_1, R_1, P_1 \rangle$ if the mappings

$$S_1 = \langle W_1, M_1, R_1, P_1 \rangle \xrightarrow{h_1} S_1' = \langle W_1', M_1', R_1', P_1' \rangle$$

$$S_2 = \langle W_2, M_2, R_2, P_2 \rangle \xrightarrow{h_2} S_2' = \langle W_2', M_2', R_2', P_2' \rangle$$

are *homomorphic,* Figure 2.3, and the mapping

$$S_1' = \langle W_1', M_1', R_1', P_1' \rangle \xrightarrow{f} S_2' = \langle W_2', M_2', R_2', P_2' \rangle$$

is *isomorphic* (one-to-one homomorphism), Figure 2.4.

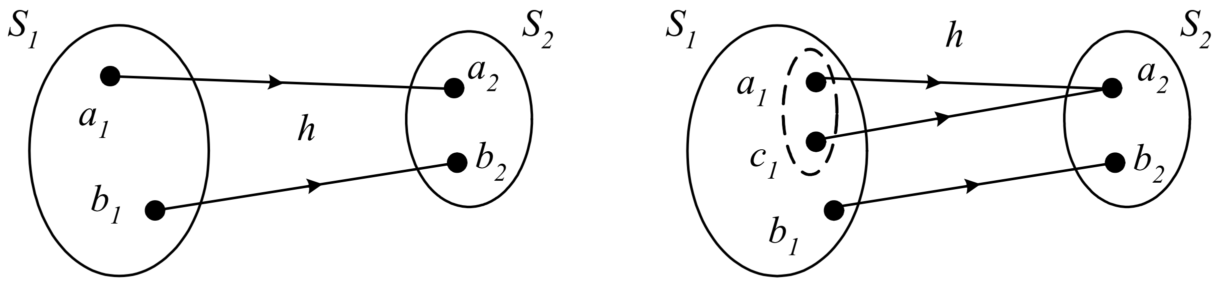

Figure 2.3. Homomorphic mapping $S_1 \rightarrow S_2$. S_1=<W_1,M_1,R_1,P_1> and S_2=<W_2,M_2,R_2,P_2> are systems. Elements a_1 and b_1 belong to M_1; elements a_2 and b_2 belong to M_2. $r_1 \in R_1$ and $r_2 \in R_2$ are relationships between elements in S_1 and S_2 respectively. $p_1 \in P_1$ and $p_2 \in P_2$ are properties such that $r_1(a_1,b_1)=p_1$ and $r_2(a_2,b_2)=p_2$. h is a mapping of S_1 on S_2 such that $h(a_1)=a_2$ and $h(b_1)=b_2$. The mapping h is homomorphic if $h(r_1(a_1,b_1)) = r_2(h(a_1),h(b_1)) = r_2(a_2,b_2)$, i.e. $r_1(a_1,b_1) = p_1$ implies $r_2(a_2,b_2) = p_2$. In other words, homomorphism preserves the relationships of S_1 in S_2. This definition allows a_1 and c_1 to be "clustered" and mapped on a_2 (e.g. when the hook and the load of a crane S_1 are represented as one material point in the crane model S_2). S_2 and the "clustered" S_1 are isomorphic

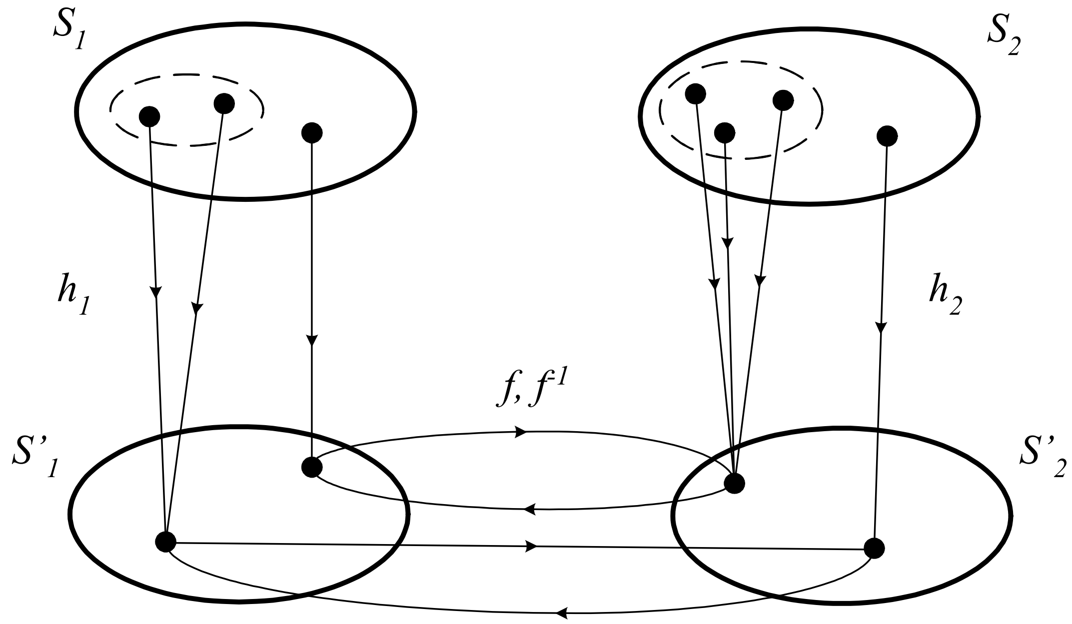

Figure 2.4. Definition of the formal model (h_1 and h_2 are homomorphic mappings, f, f^1 is an isomorphic mapping)

The meaning of this definition is that we require isomorphism only between some simplified images of S_1 and S_2. On the one hand, comparing two simplified systems $S_1^{'}$ and $S_2^{'}$ makes sense because the cardinality of a real system is usually[10] far greater than the cardinality of the corresponding model. On the other hand, there is no reason to require a model (image) to be always "simpler" than its original. For example, consider a real plant S_1 and its DVP S_2. By the above definition, the complexity of S_2 is not limited. Therefore, the DVP (VP) S_2 can describe the aspects related to the overall lifecycle which are not necessarily considered in S_1. From the viewpoint of the control system development, the homomorphic mappings h_1, h_2 have to take into account those characteristics of S_1 and S_2 which are relevant to the control loop.

Systems (more precisely, their images $S^{'}$) can be compared with respect to *the algebraic and the dynamics similarity* [Gai2001, Pod2002, Chu2003].

Two systems S_1 and S_2 are similar in the algebraic sense if $S_1 \rightarrow S_1^{'}$ and $S_2 \rightarrow S_2^{'}$ are homomorphic and $S_1^{'} \rightarrow S_2^{'}$ is isomorphic.

The dynamic similarity is investigated in the phase space by superimposing the phase

[10] The cardinalities of a real system and its physical model (analog), though, may be comparable

portraits of the compared systems[11]. The ***qualitative*** (topological) dynamic similarity is the similarity of the phase spaces structure[12]. For example, two systems are not topologically similar if in some region of the phase space one system has a stable focus and the other has a saddle point. The ***quantitative*** dynamic similarity is the closeness of the phase portraits.

Usually the mapping $S \to S'$ implies some coordinates of the system being neglected. It is important to make sure that the influence of the neglected coordinates is "insignificant". Practically it means that the system is stable in the neglected coordinates and the quantitative impact of those coordinates is small in comparison with the retained coordinates. Let us illustrate this using a gantry crane[13] example, Figure 2.5.

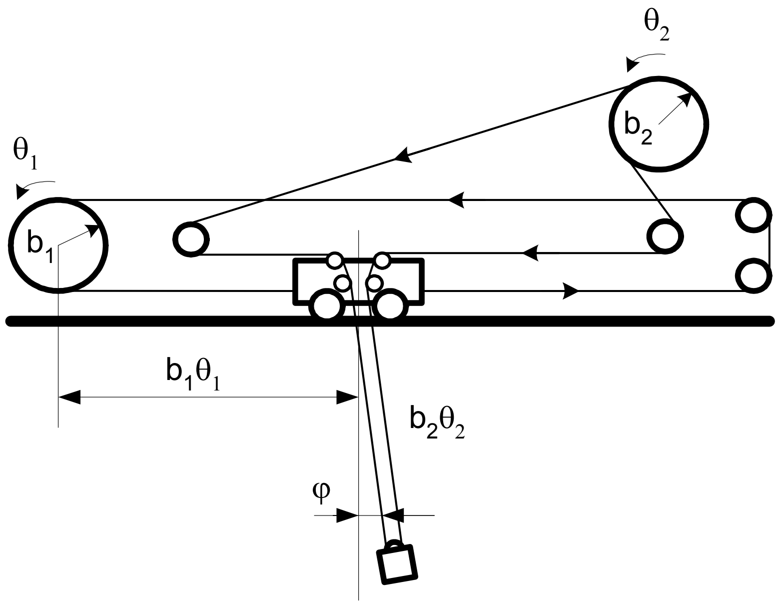

Figure 2.5. Model of a gantry crane. Degrees of freedom: {θ_1, θ_2, φ}

Suppose the DVP of the gantry crane S_2 is described in the six-dimensional space (the trolley position θ_1, the rope length θ_2, the load angular displacement φ, and their first derivatives), but S_2' corresponds to the four-dimensional space, where the nonstationary rope length is not taken into account. The rope-length coordinate of the open-loop gantry crane is unstable, i.e. the rope "unwinds" if no special measures (braking) are taken. However, the dynamics of both S_2 and S_2' in the retained coordinates is, apparently, topologically similar. Thus, S_2 and S_2' may have a stable focus in the phase planes "trolley position vs. trolley

---

[11] The dimensionality and the scale of the phase spaces should be the same

[12] The compared models have to be topologically robust in the Andronov-Pontryagin sense (see later in this section)

[13] For more detail on the gantry crane model see Chapter 5

velocity” and “angular displacement vs. angular displacement velocity”. Here, an unstable neglected coordinate has a negative impact on the quality of the model S_2' because the model is unable to capture such a significant feature of the system as instability. As a result, the discussed reduction (homomorphic mapping) $S_2 \to S_2'$ is not valid. It is impossible to find out whether the homomorphism $S_2 \to S_2'$ is valid only by means of studying the dynamics of the reduced system S_2'. One has to involve some additional “external” information about the system dynamics (Gödel’s incompleteness theorem, [Usp1982], see also Section 3.1.3) and to consider the semantic component of the problem (see below).

Depending on the number of compared processes, the dynamic similarity can be local, limit, or total [Pod2002]:

- Local similarity – the similarity of a single transient process with the given initial conditions or the similarity of a single locally stable steady-state process;
- Limit similarity – the similarity of a number of processes, which belong to a given bounded region of the phase space;
- Total similarity – the similarity of all possible processes.

Unlike similarity, *adequacy* is a more fuzzy term. “Adequate” is broadly defined as being “equal to the requirement or occasion…; fully sufficient, suitable, fit” [Ade1996]. Hence, the meaning of adequacy depends on the requirement. For example, if the dynamic behavior of two systems is compared, adequacy may be equivalent to the algebraic and dynamics similarity[14]. However, in the context of the control system development, constructing a model of the plant is just an intermediate step. Therefore, adequacy has to be interpreted with respect to the ultimate result, i.e. the control system: a control system is adequate if the functional and nonfunctional requirements are met for ***both*** the model and the ***actual*** plant.

Let us discuss this problem from the point of view of *robustness*. It is known that the engineering practice gives preferences to simple[15] models, which describe the dominant processes, and robust[16] controllers, which cover for the model uncertainties [SP1996, Ite2002].

The most general definition of robustness in the topological terms was introduced by A.A. Andronov and L.S. Pontryagin in 1937 [BNF1987]: “a system is robust if small

[14] That is how adequacy is understood in the literature on modeling, e.g. [Dyn1978, p.106]
[15] Not only “simple” in structure, but also “reliable” or robust in the topological sense
[16] In the sense of the robust stability (quality) for a given region of the model space

variations of its parameters do not change the qualitative structure of the phase space". An engineering interpretation is that a control system should be insensitive to the model uncertainties and have a sufficiently large stability margin.

Nowadays a stronger definition of the robust stability (quality) is used [SP1996]: "a control system is robust if it is stable…" (the quality criteria are met) "…for all perturbed plants about the nominal model up to the worst-case model uncertainty". This definition assesses the robustness in a given region of the model space rather than in a close proximity of the nominal model.

Note a difference between the two definitions of robustness. The topological robustness is applicable to systems in general (plants, models, control systems), while the second definition is valid for the control systems only.

The fundamental difficulty is that control systems are designed for plant ***models***, but it is the ***actual*** plant, to which the designed control system is applied. Strictly speaking, control theory cannot guarantee that the designed control system would work on the real plant even if the model seems to capture the essential behavior of the plant and the designed control system is robust. In other words, the designed control system is adequate to the actual plant if the chain "plant-model-controller-plant" is transitive, i.e. if (plant $\Rightarrow$ model) and (model $\Rightarrow$ controller) implies (plant $\Rightarrow$ controller), where "$\Rightarrow$" stands for "adequacy" [Dyn1978]. There is no general rule of insuring the transitivity except for some engineering rules of thumb, e.g. "simple model + robust controller". In a way this is a consequence of Gödel's incompleteness theorem [Usp1982]: the adequacy of the designed control system can be validated only with some external reference, ultimately, with the real plant.

An additional difficulty arises when data is employed as a basis of modeling. If a model fits the experimental data or observations, it does not imply that the model itself is correct. The data does not necessarily describe all regimes of the system. Therefore, all data-based control design methods share the same drawback that the designed control system can be assumed to be adequate only in the regimes for which the data were obtained[17] [Mod1974, Hyo2003].

The discussion can be supported by mapping the control development problem on the *semiotic system* [PY2001]:

- Syntactic level corresponds to the structure and parameters of the plant or its data-based model;

[17] In fact, the same is true for a linearized model, which describes the nonlinear model only in some specific regimes

- Semantic level is the “meaning” of the model and the development problem, context, some qualitative information, which allows to assess the applicability of the model and the designed control system;
- Pragmatic level corresponds to the control system, i.e. the system responsible for taking the control decisions.

If the semantic level is ignored, an attempt at formally transiting from the syntactic level (model) to the pragmatic level (control system) will lead to an inadequate control system.

2.4. Computer aided control system design

To respond to the tendencies outlined at the beginning of the chapter the control system development process is required to be more flexible and less consuming in terms of the time, financial, and intellectual resources. If the control system development process is inefficient, it will jeopardize the positive impact of the control system on the quality, efficiency and competitiveness of the technical system as a whole.

The general criteria and directions of improving and facilitating the processes of the technical system lifecycle are the following [MA1998]:

- Automation of routine operations;
- Rationalization by eliminating unproductive operations;
- Information integration and facilitation of information flows.

The formulated criteria and recommendations are rather universal and, therefore, can be applied to the control system development process. Let us analyze this process from the point of view of automation, rationalization and integration.

Computer-aided control system design (CACSD) systems are intended for automating the control design process. CACSD covers the following three aspects: the mathematical modeling of systems, control system analysis, and control system design [Gru2001].

In Section 2.2, it has been shown that obtaining the plant dynamics model is very difficult to formalize because this phase is strongly coupled with the physical plant. The remaining phases of the development process are quite abstract (in case of synthesis) or rely on the adequacy of the model, which is obtained at the first phase (in case of implementation).

Unlike the physical experiments, the analytical modeling can be significantly improved and intensified by means of automating the construction of the plant dynamics model.

The motivation for automating the construction of the plant dynamics model is that modern technical systems can be rather complex and contain a large number of elements of diverse nature. As the complexity of a system increases, so does the size of the equations of dynamics. Even for mechatronic systems, the number of the terms in the equations of dynamics is roughly proportional to the third order of the degrees-of-freedom (DOF) [GK2004]. As a consequence, the equations of dynamics can be derived with paper and a pen only for fairly simple systems.

Constructing and manipulating the equations of dynamics can be accomplished, for instance, using the symbolic computations, MODELICA language, and bond graphs.

The software for the symbolic computations such as Maple [Map2005], MathCad [MCAD2005], MATLAB / Symbolic Mathematics toolbox [MLAB2005] can be employed for automating the construction of the dynamics equations on the basis of, for example, the Lagrange formalism [Bel2000, GK2004, Tsa1999, ZY2000]. The symbolic computations are capable of automating only the routine mathematical operations, while the physics of the modeled system anyway has to be described by the engineer, e.g. in terms of the potential and kinetic energy functions. However, there are special methods [GK2004, Bel2000], which allow to construct dynamic equations on the basis of a kinematic graph with known dynamic parameters. In this respect the symbolic computations are indispensable for manipulating (simplifying, transforming, linearizing) the constructed equations of dynamics.

The MODELICA language [MCA2005, MEO1998] is an object-oriented language for modeling complex heterogeneous (mechanical, electrical, hydraulic etc.) physical systems. There exist the standard libraries of typical components and the tools for creating new components out of primitive ones (friction, inertia, resistance, amplifiers and many others). The primitives are internally described by ordinary differential and algebraic equations. A feature of MODELICA is that the models can be repeatedly reused in accordance with the object-oriented and component-based approaches. MODELICA is complemented with the external software for manipulating, simplifying, and simulating the composed equations. The most popular external software is MathModelica [MM2005] and Dymola [Dym2005]. The latter is able to tackle a large number of equations (up to one hundred thousand), has some visualization functionality and an interface to SIMULINK.

The bond graphs [Bon2005, KMR1990] is a graphical language for modeling heterogeneous physical systems. The basis of the bond graphs is the graphical representation of the energy flows and their transformations. The bond graphs allow to obtain an explicit dynamic model with respect to the defined input/output variables. Yet, the engineer is required to define the underlying equations that describe the interrelations between the elements of the system. Hence, the bond graphs can be regarded as a method for simplifying the construction of the overall model of the system and making this process more formal.

Once the equations of dynamics are composed or the data-based model is obtained (from the plant or its physical analog or by means of simulation), it is possible to use the software which automates the formal, ***abstract*** methods of data-processing, identification, analysis and control system design [Gru2001, RJ1996, Ste2000]. The abstraction from the physics of the real plant is made possible by means of the analytical or data-based model. The purpose of CACSD is to enable the control engineer to concentrate on designing the control law rather than programming mathematical methods used for the design. The most widely used and well-developed CACSD are MATLAB (Control System toolbox and other related toolboxes) [MLAB2005], $\mathrm{MATRIX_X}$ [MRX2005], and Scilab [SLAB2005].

In summary, the automated construction of the equations of dynamics and the CACSD systems prove to be mature and the ***automation*** dimension of the control system development process appears to be rather strong (the physical implementation of the designed control system is out of the scope of this dissertation).

By contrast, the level of the ***information integration*** between the control system development stage and other stages of the technical system lifecycle is insufficient. Today, the control system development stage is considered in isolation from both the preceding (development of the plant) and subsequent (manufacturing, operation) stages of the lifecycle. There are several difficulties related with the lack of the information integration:

1. Constructing the plant dynamics model is inefficient unless the necessary information (geometry, kinematics, dynamic parameters) is extracted directly from the VP, which has recently become the primary representation of technical systems. However, the notion of the VP has not yet been comprehended and adopted by the control engineering community;

2. There is no single reference information about the plant because the information is divided between the VP and the model of dynamics. The problem is that the development of the plant and the control system can be accomplished by different people and organizations. This hinders the modifications of the plant in the concurrent engineering environment from being with no delay reflected in the model of dynamics and vice versa;

3. The feedback from the testing, manufacturing, operation, maintenance stages of the lifecycle to the control system development stage is very limited or hampered unless all these stages share one source of information on the technical system. This feedback is important

because it makes it possible to adjust the control system according to practical experience gained at these later stages of the lifecycle.

In conclusion, despite the CACSD (including the tools for automating the construction of the equations of dynamics) systems being very important with respect to automation, it is the information integration that is the key to further radical improvements of the control system development process.

2.5. The problem formulation

Let us summarize the conclusions, which have been drawn in the preceding sections. On the one hand, the engineer confronts the fact that the DVP has become the primary source of information about a technical system. On the other hand, the integration of the control system development process into the overall technical system lifecycle is the key to raising the efficiency of the development process above the level, which is achievable solely by means of the conventional CACSD.

The consequence is that there is a need for merging DVP and the development of control systems to address both the integration and the automation dimensions of the control system development process.

Indeed, the control system development process can be integrated with other stages of the technical system lifecycle only if the ***conventional, standard*** DVP, which is employed throughout the lifecycle, is ***directly*** utilized as the information basis for obtaining the dynamic model of the plant required for the control system design. In other words, the control development process has to be adapted for the DVP/CALS/PLM lifecycle.

Automating the construction of the plant dynamic model means that the information necessary for constructing the dynamic model is ***automatically extracted from the DVP***. The DVP systems are known to be capable of automatically constructing the equations of dynamics, but this is a different question because those equations are not necessarily suitable or readily available for designing control systems.

Note that we do not attempt to automate those phases of the development process that require creativity, informal approach, unique expertise, e.g. defining and formalizing the problem, selecting specific mathematical methods, choosing model structures. Only is the routine and laborious work considered to be automatable, for example, translating the (D)VP/CALS model into the model suitable for the control system design, constructing the equations of motion etc.

Having discussed the main idea of this dissertation let us consider the obstacles, which are ***foreseen*** on the way of employing DVP for the control system development. These aspects will be also discussed at length in the subsequent chapters.

The essence of the information integration problem is in ***extracting*** the necessary dynamic model from the DVP rather than in developing a universal system representation format[18]. However, it is not trivial to "extract" the dynamic model from the conventional structural-parametric DVP. The DVP and the dynamic model have different goals and represent the dynamics information in different ways. As a result, there is some kind of an information "barrier" between the DVP and the control system development.

This dissertation aims at eliminating the barrier and adapting control system development process for the DVP/CALS lifecycle by means of developing a special method, which can be interpreted as an "interface" between the DVP and the control design methods.

The ***dynamic simulation*** of the DVP appears to be the simplest and the most universal approach to extracting the desired dynamic information from the DVP. First, all DVP systems have the dynamic simulation functionality. Second, when only the simulation data is used, there is no need to deal with the DVP system implementation details. Third, the simulation method is believed to be a natural tool for studying complex systemic phenomenological models such as the DVP [NY2001].

So, the dynamic simulation is the core of the method proposed in this dissertation. Hence, the proposed method should include the methods of modeling, reducing and transforming models, planning and conducting the virtual identification experiments, and designing control systems. The application of the conventional identification and control methods is expected to have some specifics due to the fact that it is the DVP which is used as the basis for the control system development rather than the real plant or the analytical model.

The purpose of the method is threefold. It will

1. Improve the efficiency of the control system development process in terms of increasing its flexibility and reducing the consumption of the time, financial, and intellectual resources;
2. Integrate the development process into the overall technical system lifecycle;
3. Extend the applicability of the DVP systems to the development of control systems.

---

[18] Especially as a number of universal formats already exist, for example ISO STEP [ISO2005]. This format has been already utilized for representing the ready-made dynamic models within the control development process [VHJ1996, VHJ1999]

2.6. Related work

The application of DVP to simulating the closed-loop control systems has apparently been one of the main motivations for developing the DVP systems themselves. Thus, [Ost1984] refers to modeling and simulating the controlled dynamics using DVP (e.g. ADAMS, DADS) and indicates the importance of constructing the dynamics equations by the DVP systems. There are several cases of using the data, which are generated by DVP tools (NASTRAN), in the control design process [SATTS1993, YC1993]. [KO1994] describes the anti-lock brake control system modeling and simulation with DVP (DADS). [HC1999] presents an approach for using DVP for control design to reduce the design cycle and costs. A very detailed paper [WB2000] concentrates on modeling (with IDEAS), simulation and model validation of an industrial paper machine and proposes to employ the model for tuning the existing controller of the machine. [RJ2004] suggests a co-simulation application based on DVP (ADAMS) and CACSD (MATLAB). In all these applications the DVPs are constructed for the purpose of dynamic simulation of specific systems and there is no mention of the information integration issues.

Furthermore, all major developers of the DVP systems declare the simulation of control systems as an important practical application of their products e.g. [LMS2001, MSC2005, Del2005].

However, the idea of employing DVP for the control system development or integrating the control system development with DVP has emerged in the past decade. Thus, papers [VHJ1996, VHJ1999] propose a unified information model architecture for the information exchange between various CACSD tools. It appears that the unified model, which is described in those papers, is equivalent to the VP concept. Both papers explore the information dimension of the problem and elaborate on the integration of the control system development lifecycle by means of the ISO STEP standard [ISO2005]. The papers, though, do not concern the transformation of the existing product STEP model (geometry, kinematics, dynamic parameters) into the model, which is suitable for the control system design.

The idea of a broader integration is suggested in [Gru2001], i.e. the development of the information model for CACSD in the ISO STEP framework linking the control system development process with the remaining stages of the technical system lifecycle.

In [UBMS2000], it is proposed a unified architecture for the computer-aided design of mechatronic systems, which involves the joint application of such tools as the (D)VP systems, the symbolic computation systems, CACSD. Yet, little attention is paid to employing the DVP directly for the control system design.

The application of DVP to the control system development appears to be a natural consequence of the DVP and CACSD evolution. For example, in [LMS2001] there is a reference to a practical project, where the LMS DVP system had been employed as a basis for the control system development. There are two more papers providing further insight into that project [BG2000a,b].

The purpose of the project described in [LMS2001, BG2000a,b] was to design the control system for a large robotic manipulator. To speed up the design process a DVP approach was adopted. The mechanical, electrical, hydraulic phenomena and the elasticity of the manipulator were taken into account and modeled in the LMS DVP system. The control system was designed on the basis of the simulation data generated by the DVP. Finally, the developed control system was evaluated using the DVP. However, the papers do not provide an exhaustive description of how the control system development process was implemented.

To our knowledge, [LMS2001, BG2000a,b] are the first publications describing the project which not only utilizes DVP for analysis, but also actively employs DVP for the purpose of the control system development.

Article [Fed2004] refers to a simple case of studying the dynamics of a virtual car implemented in the “Need for Speed 3” computer game. It is interesting that the virtual car is considered as a black-box phenomenological model with no explicit reference to the underlying equations of dynamics. This approach complies with our understanding that the virtual prototypes (models) have to be treated as real physical objects.

During the preparation of this dissertation, the MathWorks released SimMechanics, an extension of Simulink (in 2004 according to [MLAB2005]). SimMechanics practically implements the idea which lies behind this dissertation. The SimMechanics software is capable of modeling mechanical systems (the initial information is the kinematic graph and the dynamic parameters of the plant; the equations of dynamics are composed automatically), simulating their dynamic behavior, and facilitating the control system design (there are tools for extracting a linearized model of the plant).

At present SimMechanics is able to import only the SolidWorks CAD data (the models can be constructed form scratch in the Simulink environment). Moreover, SimMechanics does not assume any integration of the control system development process into the overall technical system lifecycle. As a consequence, SimMechanics (in conjunction with the Virtual Reality toolbox) cannot be classified as a true DVP system because its information integration dimension seems to be quite weak.

In contrast to the SimMechanics approach, this dissertation offers an ***open*** solution to the development of control systems. First, the proposed method is vendor-independent; it can be implemented by means of a multitude of the DVP and CACSD tools available in the market. Second, depending on the nature of the plant, a suitable DVP system or dynamic engine can be employed. So, the proposed method is virtually invariant to the nature of the plant. Third, the proposed method heavily exploits the idea of the information integration between the control system development stage and other stages of the technical system lifecycle.

In conclusion, the underlying idea of using the DVP for the control system development has been independently proposed by a number of authors, which have considered the problem from various viewpoints. Furthermore, the recent release of SimMechnics by such a major company as the MathWorks shows that the research and development in this domain is very active.

Unlike the above-described general ideas, methodologies, specific applications or software, this dissertation provides:

1. A ***systematic method*** employing the ***conventional*** DVP for the ***development*** of control systems and ***integration*** of the development process into the ***overall lifecycle***;
2. An ***implementation and adaptation of the method*** to a fairly wide ***class*** of plants.

2.7. Summary

1. In this chapter it has been analyzed that nowadays the control engineer confronts the fact of the DVP is becoming the primary source of information about the technical system. It compels the engineer to develop control systems on the DVP basis extracting the necessary dynamic models directly from the DVP.

2. The automation dimension of the control system development is addressed by CACSD, whereas the information integration dimension is less supported by the computer-aided tools. However, it is the information integration of the control system development process into the overall technical system lifecycle that is likely to raise the efficiency of the development process above the level achievable solely by means of the conventional CACSD. DVP turns out to be the key to this integration.

3. An information "barrier" between the DVP and the control system development has been revealed. The essence of the problem is that it is difficult to "extract" the dynamic

model, which is suitable for the control system design, from the conventional structural-parametric DVP. The DVP and the dynamic model have different goals and represent the dynamics information in different ways.

4. The need for a method ("interface"), which would eliminate this barrier and adapt the control system development to the DVP/CALS lifecycle, has been revealed and justified in this chapter.

5. The proposed method has been compared with the related works. It has been found that the domain of DVP and the control system development had been rather carefully explored by a number of researches, e.g. [UBMS2000, LMS2001, BG2000a,b]. However, the main scientific strength and novelty of this dissertation is that it provides (1) the systematic method employing the conventional DVP for the development of control systems and integration of the development process into the overall lifecycle and (2) an implementation and adaptation of the method for a class of plants.

CHAPTER 3. THE DVP METHOD FOR THE DEVELOPMENT OF CONTROL SYSTEMS: DESCRIPTION AND ANALYSIS

In this chapter the theoretical and methodological foundation of the proposed development method is elaborated. The main concept of the method, the description of its principal stages, the analysis of the efficiency and the practical applicability of the method and the adequacy of the developed control system are presented.

3.1. Description of the method

3.1.1. Conceptual model of the method

The main idea of the proposed development method is illustrated in Figure 3.1 (the differences between Figure 2.2 and Figure 3.1 are shown in dashed lines). The method comprises the following three stages:

1. Representing the plant by its dynamic virtual prototype (DVP);
2. Simulating the DVP and generating the data-based model of the plant;
3. Designing the control system using the generated data-based model.

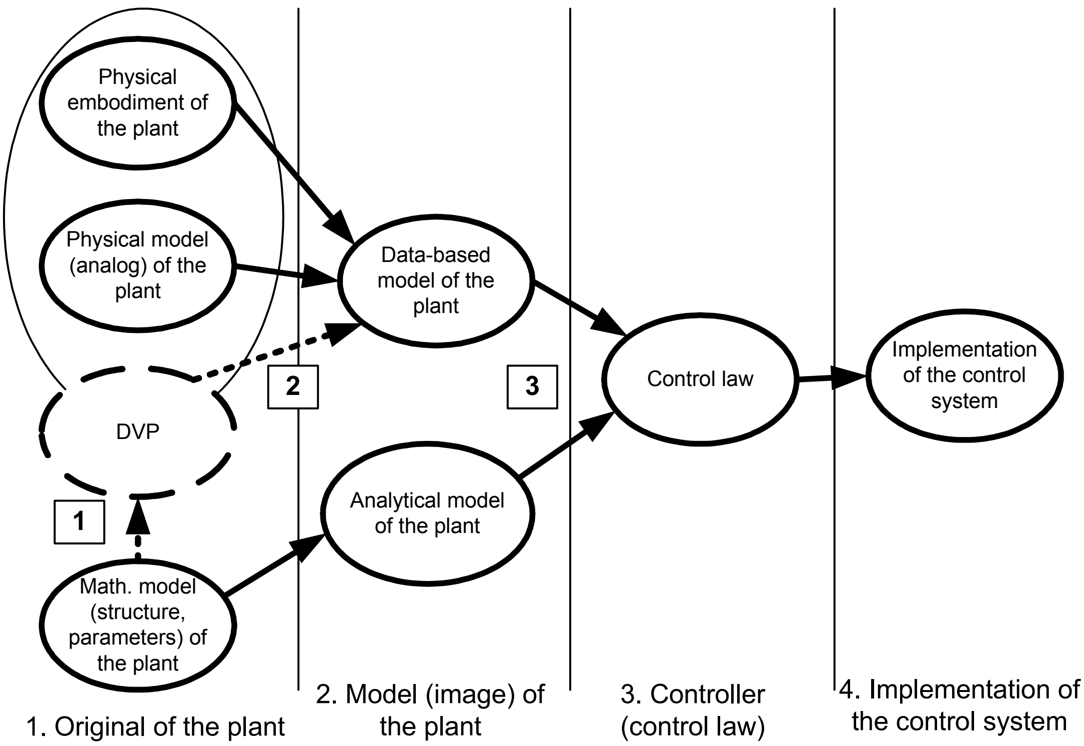

Figure 3.1. Conceptual model of the DVP control system development method (compare with Figure 2.2)

The core of the proposed method is in substituting the simulation of the DVP[1] for an analytical model or a physical embodiment (physical model) of the plant. Moreover, the method enables to merge the DVP/CALS systems with the control system development. As a result, the theory and practice of the control system development benefit from the DVP/CALS systems, which, in turn, extend their application domain.

The proposed development method can be represented as a procedure that has an input and output and comprises the previously-mentioned three stages, Figure 3.2.

The plant DVP is handled by a DVP system (e.g. IGRIP [Del2005]), whereas the manipulations with data-based model, the analytical model identification and the control system design are supported by CACSD (e.g. MATLAB [MLAB2005]). The implementation issues will be described at length in Section 4.3.

3.1.2. The input and output of the method

The input of the method, Figure 3.2, is the information on the plant in the form of the technical documentation, drawings and specifications. Depending on availability, any existing CALS models or (D)VPs, models of plant's former generations, analytical relationships and expert knowledge can be exploited. The possibility of (re)using the existing CALS models is a feature of the method. Despite the contemporary CALS systems being mainly confined to the geometric properties of the plant, even a ready-made geometric model of the plant significantly reduces the laboriousness of constructing the DVP.

The input of the method should contain information on the structure and parameters of the plant, which permits to define the input and output of the plant, as well as its input-output mapping dynamic operator. This information has to correspond with the system complexity level of the control loop [PY2001, LP1999]. Moreover, some qualitative information is needed to define the semantics (meaning) and to elucidate the applicability area and conditions of the constructed DVP [PY2001]. On the whole, approximately equal amount of the input information is needed for constructing the DVP and for the analytical modeling.

Furthermore, along with the DVP, the control problem has to be formulated in terms of the functional and nonfunctional requirements to the control system. The functional requirements refer to the input-output transformation of information (signals), e.g. manipulating the input of the plant in order to stabilize it or to insure the desired dynamic performance.

---

[1] The DVP system automatically constructs the dynamic equations of the plant

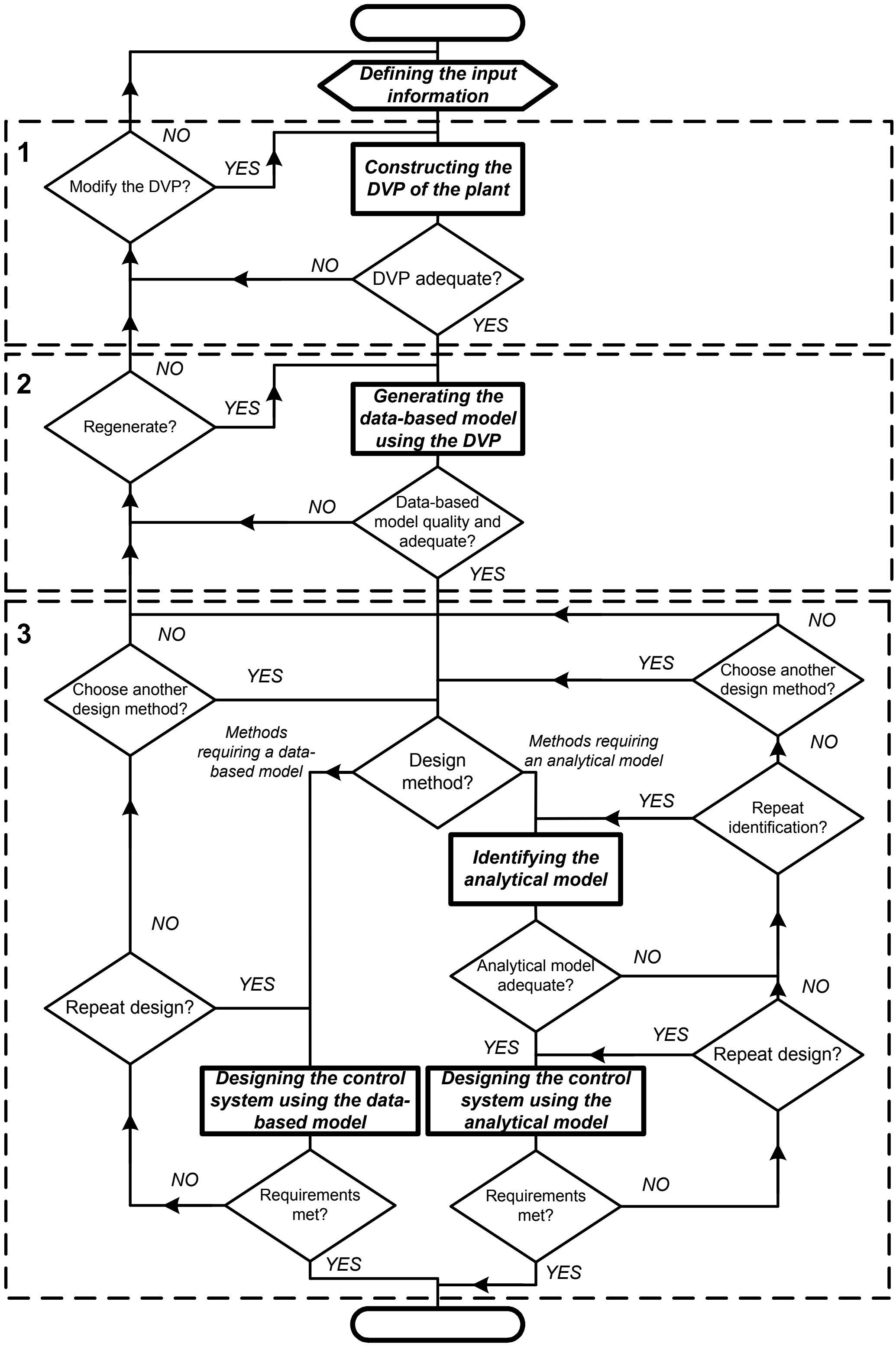

Figure 3.2. Main stages of the DVP control system development method

The non-functional requirements refer to the quality of this transformation, i.e. settling time, oscillateness, static error, maximum control energy, and robustness to plant uncertainties and external disturbances [SP1996, Bur2001].

The output of the method is the developed control system, more precisely, the control law, which solves the control problem and satisfies the defined technical requirements. The implementation issues and the physical embodiment of the control system are left beyond the scope of this dissertation.

3.1.3. Stage 1: constructing the DVP

If the plant's DVP does not exist, it has to be constructed on the basis of the method input (Section 3.1.2). This stage comprises the following four steps:

1. Selecting an appropriate DVP system;
2. Modeling the plant in the DVP system;
3. Composing the equations of dynamics using the functionality of the DVP system;
4. Analyzing the adequacy of the constructed DVP of the plant.

<u>*The DVP system selection*</u> depends on

- The class of the dominant physical phenomena, which are the most significant and relevant for describing the dynamics of the plant related to the control loop;
- The scale of the modeled phenomena;
- The system complexity level.

If the relevant physical phenomena are not supported by the selected DVP system, those phenomena have to be defined analytically. For instance, if a DVP system is capable of modeling mechanical systems (e.g. IGRIP), then the mechanical subsystem of the plant can be modeled directly. In other words it is sufficient to define the structure and parameters of the plant as the input information. However, the electro-mechanical (electric motors, sensors and transducers of the mechanical quantities into the electrical ones), aerodynamic (aerodynamic drag) and other types of phenomena have to be described analytically. The direct modeling of several physical phenomenon types becomes possible when such DVP systems as ADAMS and LMS/VirtualLab are employed. A condition for the efficient application of the proposed development method is that the DVP system has to be capable of direct modeling those elements of the plant which pose the primary difficulties for being modeled analytically.

Modeling of the micromechanical systems is an example when it is crucial to coordinate the scale of the plant phenomena and the type of the DVP system. When the physical dimensions of the micromechanical systems become so small that the surface tension force and other micro scale phenomena start taking prominence, it should be taken into consideration in an appropriate DVP system [COM2005, FEM2002].

Finally, a consideration has to be given to the system complexity level. The proposed development method suggests reusing all available CALS models of the plant for constructing the DVP. Yet, some CALS models are known to contain information on up to several thousands or even millions of parts[2]. It is not always desirable or technically feasible to model the dynamics of such a system at the micro level, i.e. taking into account the contribution of all individual parts[3]. Moreover, in accordance with the engineering practice, the quality of the simulation does not improve if the real parameters of the system are unknown and their values are hypothesized [Fun2001][4]. As a result, should the CALS model be made use of, it has to be converted to such a model that describes plant's relevant dynamics on the whole, at the macro level, and, paraphrasing [PS1985], retains certain "robustness" when an approximated model substitutes for the complete one.

Leaving out the development of macro models, note that usually it is difficult to construct (or to automate the construction) a macro model on the basis of a micro model mainly because of the emergent phenomena, i.e. the qualitative changes in the system implied by the transition from the micro to the macro level[5] [Hyo2003, DK2001, Kol1994, Sam2001]. Hence, in this dissertation it is assumed that the input information about the dynamics describes the plant at the macro level. However, subject to the necessary qualitative information being available, it may be possible to obtain a macro model of the plant by means of (1) aggregating the individual components of the plant into larger blocks, (2) considering the plant as a whole at a high level of abstraction, (3) idealizing and (4) neglecting some irrelevant phenomena. For instance, the trolley of a gantry crane (Figure 2.5) may be represented as a material point.

When discussing micro and macro models, we refer to the system complexity of L. von Bertalanffy, who proposed the following three levels of complexity [Ber1968]:

1. "Organized simplicity";
2. "Disorganized complexity";
3. "Organized complexity".

---

[2] Airbus A380 digital mockup (VP) is known to contain four million parts [ML2005]

[3] There are a number of examples of the micro level modeling, e.g. the supercomputer modeling of floating objects with the explicit use of Navier-Stokes equations and the finite-element method [AA2002] and the full-scale test of a car crash [Gen2004]

[4] Despite this conclusion has been drawn in regard to the control systems of the high-precision missiles, the conclusion seems to be general enough for being applicable to any technical systems

[5] In fact, it is similar to one of dialectics laws: "the transformation of quantity into quality and vice versa" [Eng1883]

A model at the micro level corresponds to the first level. As the dimension of the system increases towards the disorganized complexity, the analysis and synthesis tools of the first level become inadequate. At some point a new order emerges when the disorganized complexity turns to the organized complexity, or the macro model.

For the sake of clarity, henceforth in this subsection we will primarily consider ***mechatronic*** plants [Ise2005]. However, most of the material is valid not only for IGRIP, but also for the DVP systems on the whole. As the DVP of the plant becomes a substitute of the physical plant, it makes the remaining stages of the proposed method ***invariant*** to the physical nature of the plant.

The DVP level of details[6] depends on the modeling goals, which include [Del2005, LMS2001, MSC2005]:

1. ***Modeling geometry:*** modeling the appearance and the ergonomics of the system, checking its dimensions etc. Modeling geometry is one of the necessary phases that precede modeling kinematics and dynamics. So, it makes sense to reuse the geometric model repeatedly and to accomplish this phase in the CAD environment, which is compatible with the environments of the subsequent phases of modeling. For example, if IGRIP/V5 is used, the geometric modeling has to be performed in the generic CATIA CAD [Das2005];
2. ***Modeling kinematics:*** modeling the interaction between the elements of the system without taking into account their mass, inertia, friction or other dynamic parameters. This type of modeling is exploited for solving the direct and inverse problems of kinematics, detecting collisions, determining the work space, and off-line programming;
3. ***Modeling dynamics:*** modeling the interaction between the elements of the system with taking into account such parameters as mass, inertia, friction, elasticity etc. to solve the direct problem of dynamics, i.e. to analyze the realistic trajectory of the system (command tracking). In some cases the inverse dynamics problem can be also addressed;
4. ***Modeling manufacturing, servicing, recycling:*** modeling the co-ordination of a number of devices/systems to test and to optimize engineering procedures and processes, e.g. welding, cutting, painting, automated and manual assembling and servicing etc.

In the first case (modeling geometry), the emphasis is placed on visualization; in the third one, visualization is not very important and a simple wire (dynamic) model is sufficient; whereas in the fourth case, the kinematic model of a human being might be needed (for example, V5/Ergonomics [Das2005]). This idea is illustrated in Figure 3.3.

---

[6] More precisely, the level of details of the models, which are included into the DVP

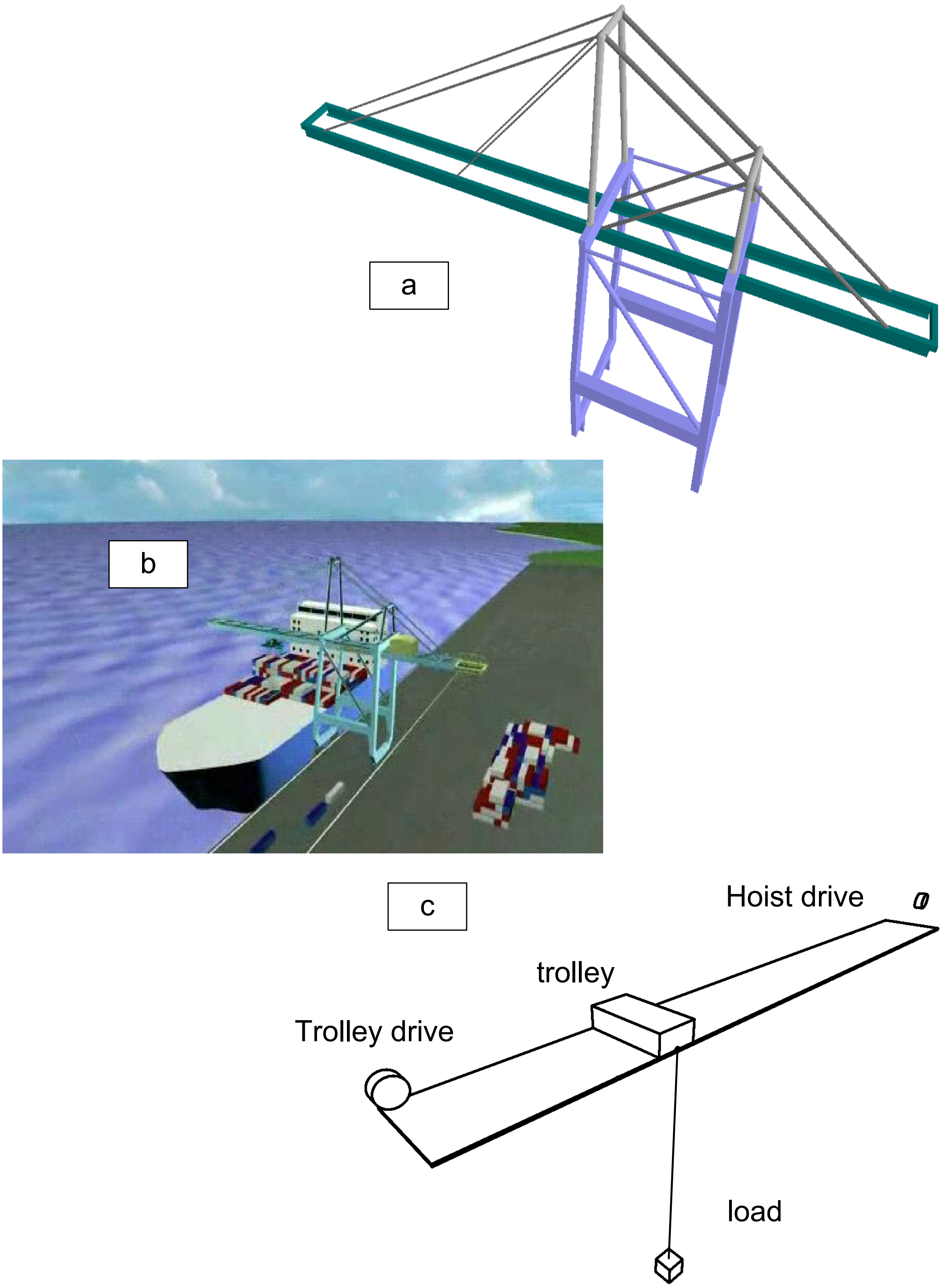

Figure 3.3. Models comprising the DVP and their level of details. Models (a) and (b) are aimed at visualization. Model (c) is a simplified wire model for modeling kinematics and dynamics

When the information about the structure and parameters of the plant is available, constructing the DVP resembles assembling the system using technical drawings or playing a LEGO game [LEG2005]. It is possible and desirable to take advantage of the available CALS models and to reuse the existing elements of the plant DVP, for instance, those constructed for the previous generations of the plant. By doing so, the experience of the development ***accumulates*** in the DVP, while the applicability of the CALS systems extends to the dynamic modeling and, further, to the control system development.

<u>*The process of modeling the plant in the DVP system*</u> consists of the following phases, Figures 3.4 – 3.7:

1. ***Constructing the 3D model of geometry.*** It is sufficient to define simplified wire models of the plant in scale 1:1 (Figure 3.4). The model of geometry is built using graphical primitives (cuboids, cylinders etc.), standard blocks and operations (union, difference, intersection and so on).

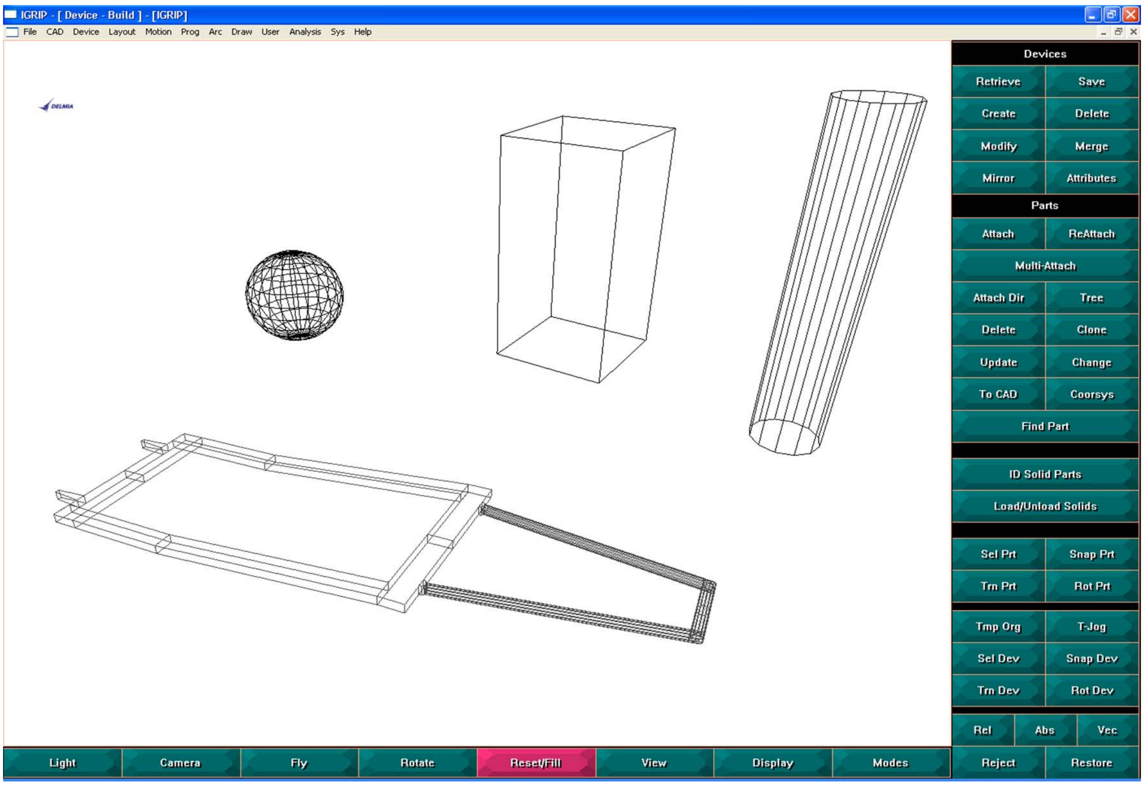

Figure 3.4. Constructing the model of geometry. Graphical primitives (at the top: sphere, cuboid, cylinder) and a combined object (at the bottom: a support structure of a gantry crane) are shown in IGRIP window

2. ***Constructing the model of kinematics*** using the information about the geometry of the individual components of the plant, their positional relationships, and the kinematic graph.

The *kinematic graph*[7] *G=<V,E>*, unlike the kinematic schemes [Tsa1999, ZY2000], schematically represents the topology of the system kinematics, while abstracting from the technical implementation details, Figure 3.5. The kinematic graph contains only the minimum necessary information. The nodes *V* of the undirected graph denote the links and its edges *E* correspond to the joints (kinematic pairs). Each edge has either a translational or a rotational degree of freedom (DOF), i.e. an edge has only one DOF. If a joint has several DOFs, it has to be represented by several edges between the corresponding two nodes. Usually it is possible to define joints of the zero through the sixth class[8]. One of the graph nodes should indicate the fixed base of the mechanism.

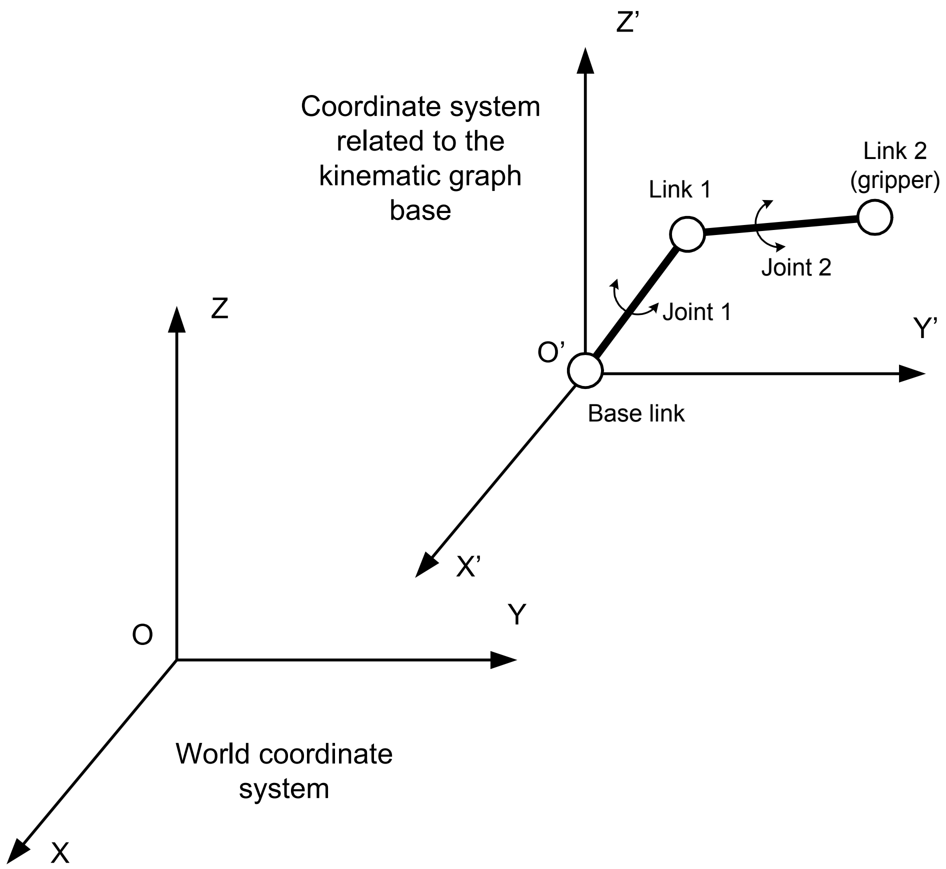

Figure 3.5. Constructing the model of kinematics. The kinematic graph

Assume the geometric models of the system parts are available. To define the kinematics of the DVP the following steps have to be performed:

- Indicate the base node (a fixed or "parent" part of the system);

[7] The description of the kinematic graph can be found in IGRIP's user manual [Del2005]. The functional joints are also implemented in IGRIP, but the notion of the functional kinematic graph has been introduced in this dissertation (see later in this section)
[8] The class of a joint (kinematic pair) is the number of the imposed kinematical constraints

- Attach the adjacent links to the base by defining the type of the joints, e.g. a translational, a rotational joint, or a combination of these two types up to six DOFs;
- Sequentially apply the second step to all links and joints of the system.

If there is a closed kinematic chain, the graph will include a circuit. To define a closed chain one has to indicate the active (driving), closing, and passive (driven) joints, Figure 3.6.

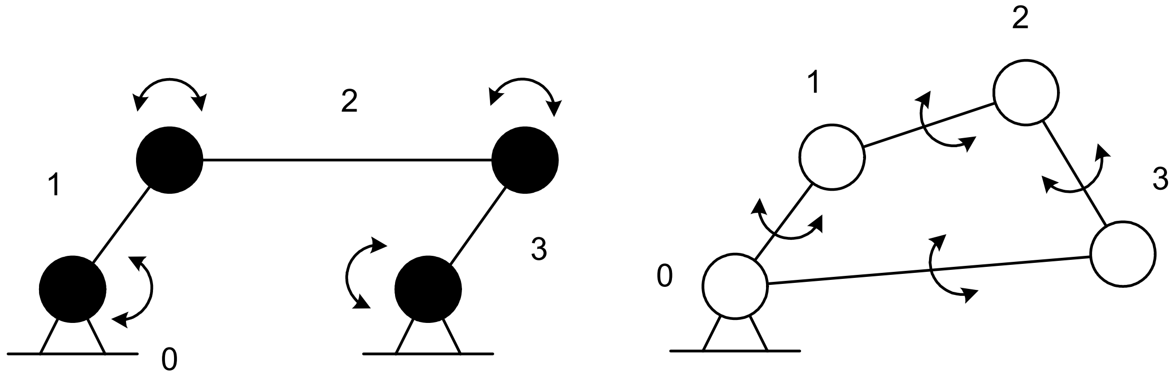

Figure 3.6. Kinematic graph of a closed kinematic chain. The layout of the mechanism is on the left (the left bottom joint is active, the right bottom one is closing) and the corresponding kinematic graph is on the right. The links of the mechanism are represented by the graph nodes, which are numbered starting from the fixed base (zero). The mechanism has four joints, but there is only one degree of freedom

Furthermore, the number of DOFs may be reduced if there are some “functionally” dependent (driven) joints, i.e. the position of a driven joint is a known function of the driving one. For example, the drive motor of a winch is a driving rotational joint, whereas the length of the hoist rope z is a driven translational joint, which is calculated as the angular position of the motor joint θ multiplied by the reduction ratio R, Figure 3.7. Using the functionally dependent joints may substantially reduce the complexity of the kinematic modeling. Thus, instead of meticulous modeling[9] of the motor, drum, reduction gear, driving ropes and sheaves, one can simplify the model by aggregating the above-mentioned elements into a simple functional relationship between the motor and the rope length.

3. ***Defining the dynamic parameters of the plant.*** Either the center of mass and the inertia matrix, or link’s mass, or the material, of which the link is made (aluminum, steel etc.), has to be defined for each link. In the latter two cases the center of mass and the inertia matrix will be calculated automatically by the DVP system if the geometry of the link is known and the density of the material is uniform.

---

[9] Unless it is not a goal of modeling

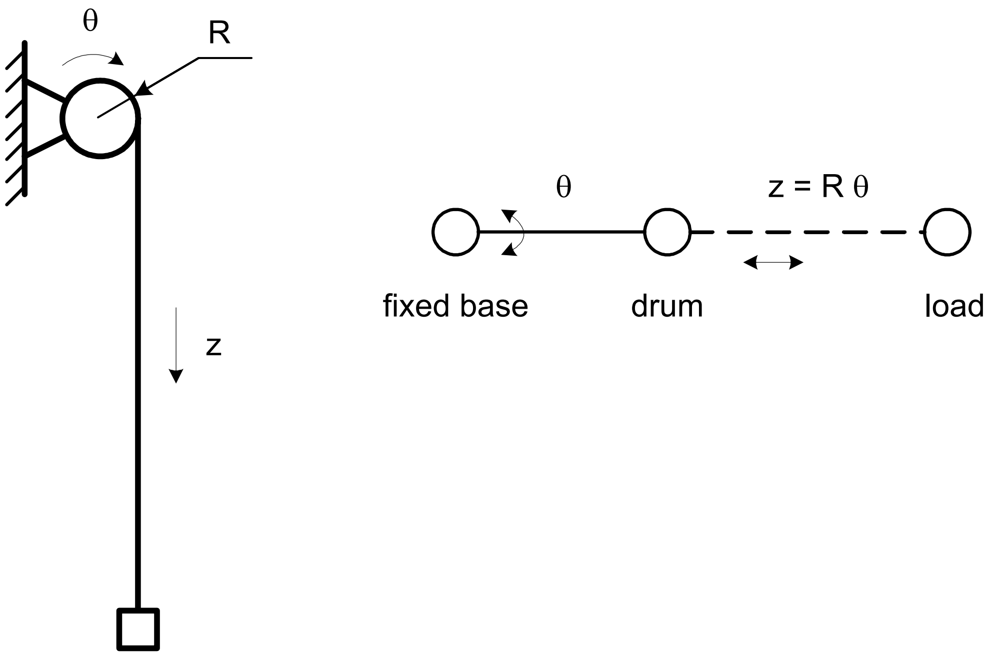

Figure 3.7. Kinematic graph of a functionally dependent joint. θ is the active, driving, joint (solid line), z is the passive, driven, one (dashed line). The mechanism has two kinematic pairs (base-drum and drum-load) denoted by two edges, but there is only one DOF

Every link can be assigned some Coulomb or viscous friction (and, therefore, e.g. the air drag) and the spring constant. The presence and the direction of the gravity force have to be taken into consideration. Also, it is possible to define external forces or torques applied to the individual links of the system, for example, the torque of the drive motors or the air drag. The nonstationary external forces and torques should be described in the simulation program (GSL in IGRIP). Finally, it is necessary to specify the initial translational and rotational positions and velocities of the joints.

Having modeled the plant in the DVP system, let us discuss the next step of the DVP construction stage: *composing the equations of dynamics using the functionality of the DVP system.*

The DVP system automatically composes the equations that describe the dynamics of the plant using the information about its structure (the geometry and the kinematic graph) and dynamic parameters. The mathematical basis for the mechanical systems is the formalism of the Lagrange equations [Tsa1999, Bel2000]. Also, there are other methods of composing the equations of motion. A comprehensive survey of methods is given in [Bel2000]; [GK2004] presents a general and computationally efficient approach to constructing the equations of motion for the systems with the tree-like kinematic graphs.

The equations, which are composed by the DVP system, ***cannot be explicitly utilized by the user for designing the control system***.

- As a rule, the equations contain the numerical values of the parameters. Hence, the equations have to be solved by means of the numerical methods (the symbolic computations are not truly efficient here);
- The equations usually contain nonlinear terms and may include non-stationary external forces, which are arbitrarily defined by the user in the simulation program;
- The composed equations are rarely made available (accessible) to the user by the DVP system developers.

As a result, the only practical way to make use of the composed equations is ***to simulate*** them in the DVP system and ***to obtain the data-based dynamic model of the plant*** (see Section 3.1.4)[10].

The final step of the DVP construction stage is <u>*the adequacy analysis of the constructed plant DVP*</u>.

The DVP is adequate if it ***captures*** the relevant dynamics of the plant (algebraic and dynamic similarity, see Section 2.3), it ***is consistent*** with the existing information about the plant and it ***is robust*** to the parametric and nonparametric uncertainties.

The adequacy of the DVP is a delicate issue. Unlike other stages of the method, the DVP construction is strongly coupled with the real plant. Referring to K. Gödel's incompleteness theorem [Usp1982] and the formula "practice is the criterion of truth", the ultimate adequacy test is the comprehensive comparison of the DVP with the actual plant (in other words, it is impossible to prove the adequacy of the model not having other references than the model itself). However, the physical embodiment of the plant is not always available to the engineer and there is no other solution than to compare the DVP with all *a priori* independent relevant information. This information may include some existing analytical and physical models, former generations of the plant, expert knowledge and any qualitative information that refers the semantics of the DVP and allows to assess the validity of the

---

[10] Simulink/SimMechanics of The MathWorks takes a different approach. In order to extract a linear model out of a Simulink/ SimMechanics model of the plant, the `linmod` function is employed. It linearizes the model block-by-block about a stationary point. The result is a linearized dynamic model of the plant in the state-space form with the numerical values of the elements [MLAB2005]. Note that in general the method proposed in this dissertation is not limited by the linearizable plants. The simulation approach allows to linearize the model about a trajectory rather than about a stationary point. Also, unlike The MathWorks' approach, it is capable of tackling the user-defined arbitrary external torques and forces

model approximations. In fact, the gradual introduction of the changes accumulates in the DVP. This makes it easier to provide the adequacy of the subsequent generations of the DVP.

Another important problem is that the DVP has to be appropriate for the control system design rather than for a mere representation of the system dynamics. Therefore, the adequacy analysis in the context of the proposed development method also pays attention to the preliminary analysis of the plant. It must be analyzed which phenomena have to be taken into account and which ones have to be neglected or simplified and vice versa ***with respect to the control system design***. For instance, a phenomenon (e.g. backlash of the actuator) being insignificant for the reasonable representation of the open-loop plant dynamics, may play a crucial role for the control system design.

3.1.4. Stage 2: generating the data-based dynamic model

As it has already been presented in the previous section, the equations of dynamics, which are composed by the DVP system, cannot be directly utilized for designing the control system. A solution is to simulate the DVP of the plant in the DVP system and to obtain the data-based (input-output) dynamic model of the plant in the desired regimes.

The DVP of the plant has to be properly excited in order to obtain the representative and rich data-based model, which contains the information on the dynamic behavior of the plant in the operational regimes of interest. The data-based model is required to be ***adequate*** in these regimes, i.e. the dynamics of the DVP and the dynamics of the data-based model have to be qualitatively and quantitatively similar (see Section 2.3).

On the one hand, an advantage of the proposed development method is that it allows to conduct a more thorough and comprehensive identification experiment and to obtain more data than if a physical embodiment of the plant were used. Indeed, the experiment with the DVP imposes almost no restrictions on the input signal. The virtual input signal may be unrealizable (in terms of its amplitude, shape, high-frequency spectrum) in the experiment with the real plant. Moreover, it is possible to experiment with marginally stable and unstable plant DVPs.

On the other hand, the plant DVP does not contain information on those phenomena that are not taken into account at the DVP construction stage. As a consequence, the data-based model generated with the DVP is not as rich as the model obtained with the real plant. Furthermore, it is infeasible to design and to test a control system for the regimes unaccounted for in the data-based model[11]. The latter issue is a limitation of the DVP control system development method as well as ***any*** data-based design method [Hyo2003].

Unlike adequacy, which addresses the dynamic similarity, the quality of the data-based model refers to the subsequent control system design. The quality of the data-based model means the degree of the model suitability for the control design methods, which are explicitly or implicitly founded on the parametric identification. For linear systems there exist some *formal* criteria for assessing the quality of the data-based model, see Section 4.2.1.1 for more detail. However, there are no formal criteria for a general type of plant. There is a rule of thumb that the data-based model has to be sufficiently "rich" to permit a reliable parametric identification of the plant.

The quality of the data-based model depends on the existence of the plant internal feedback loops. As those loops result in the dependent output and input of the plant, special methods have to be applied to avoid biased identification [Lju1999, Lju1996].

Maintaining the (open-loop) plant in the proximity of a given work region, in which it is important to have an adequate data-based model, is one of the necessary conditions of obtaining a valid data-based model. This condition is satisfied when the plant is stable and the stationary point or a stable limit cycle (in general, a region of attraction) encloses the work region. If this is the case, a bounded input signal, which does not force the plant to leave the work region, produces a bounded output (BIBO stability [Zak2003]). The resulting synchronized input-output time series will describe the dynamic behavior of the plant in the work region in question.

From the point of view of practice, it is desirable to extend the applicability of the method on some unstable plants not limiting the development method by stable plants.

The phase trajectory of an unstable plant diverges slowly when the largest positive Lyapunov's exponential is close to zero [Chu2003]. Hence, the phase trajectory does not leave the work region if the initial state of the plant is within the region and the simulation length is sufficiently short. The trajectory divergence rate can be assessed by means of the direct simulation. In some cases, when relevant *a priori* qualitative information is available, it is possible to find an appropriate input signal, which would stimulate the plant to remain in the work region.

Let us demonstrate this idea using the gantry crane example, Figure 2.5. The hoist DOF of the gantry crane θ_2 is unstable. However, it is possible to maintain the load in the

[11] This may happen when the DVP either does not contain the necessary information or does contain it, but the identification experiment fails to extract it

vicinity of the work region for a sufficiently long time and to obtain the necessary amount of data when the gravity force is compensated by introducing a constant balancing torque, which is a known function of the container mass, and the zero-mean input signal drives the load symmetrically with respect to the reference load length. Let us now consider the trolley drive DOF θ_l, which is marginally stable. As the dynamics of the crane is invariant to the trolley position, the input signal does not necessarily have to maintain the trolley around some work point. This property simplifies the shape of the input exciting the trolley. However, the crane dynamics does depend on the length of the load making the data nonlinear. Therefore, the hoist excitation input signal should be symmetric with respect to the reference load length.

Generally, it is unlikely to find a quasi-stabilizing input, which stabilizes the unstable plant without introducing a feedback loop as it is the only practical method of stabilizing unstable plants [SP1996, Toi2000]. Usually an analytical model of the plant is required to design the feedback control. Most likely there is no reliable stabilization method that does not require an analytical model of the plant. Yet, if there exists some *a priori* expert information, which is relevant to the plant stabilization problem, it may make sense to employ the fuzzy controllers to formalize the available qualitative and quantitative information in the form of fuzzy rules [Zak2003, Fuz1999, Koi2000, HMB1993]. If a feedback stabilizer is involved, some special measures have to be taken to avoid biased identification results.

3.1.5. Stage 3: designing the control system

Designing control systems strongly depends on the mathematical description of the plant (linear vs. nonlinear, stationary vs. nonstationary etc.). Since there are no “universal” control methods, some ***a priori*** information is needed to determine the class of the plant and to select the compatible control system design methods. In this section we will consider some general aspects of the control design in the framework of the proposed DVP development method, whereas the specific methods, which are applied to a selected class of plants, will be discussed in Chapter 4.

In accordance with the proposed development method, the control system (control law) is designed on the basis of the plant dynamic data-based model, which is generated in the DVP system. The applicable design methods can be classified into the following two groups:

1. The first group involves the analytical model of the plant. It is necessary to identify the analytical model using the data-based model prior to the control design;
2. The second group directly employs the data-based model of the plant (in fact, the methods of this group implicitly make use of identification procedures).

Note that this dissertation does not aim at proposing any new control system design methods. On the contrary, the goal is to produce the ***synergy*** of the DVP concept and the ***existing*** control design and identification methods.

On the whole, the robustness of the designed control system is one of the most significant problems of the proposed development method. Thus, an estimate of the model uncertainty set obtained with the plant DVP is likely to be considerably different from the estimate based on the real plant. As the DVP cannot describe those dynamic features that are not taken into account by the DVP developer, the DVP-based uncertainty set estimates may be deceptive or false. This fact, for example, limits the applicability of the data-based iterative identification and control methods [Ite2002, TS1999]. Furthermore, the classical methods of the robust control design require the structure of the uncertainty set to be defined, i.e. how the plant uncertainties affect its model [SP1996, DFT1990]. Generally, the proposed development method is not capable of providing this information because a single data-based model describes the plant dynamic behavior just in one given work regime[12]. The key to a solution may be using multiple data-based models that would describe the plant on a set of work regimes and parameters (see Section 4.2.1.2).

Referring to Figure 3.2, the adequacy of the analytical model is assessed in the same way as the adequacy of the data-based model (see Section 3.1.6). The fact that the designed control system meets the non-functional requirements (settling time, oscillateness, static error, and robustness to the plant uncertainties) is to be checked by means of the direct simulation both in the CACSD and the DVP system[13].

Since the robustness of the designed control system cannot be always guaranteed, a special attention has to be given to the exhaustive simulation and validation of the designed control system in various work regimes and with a variety of parameter values in order to assess and to validate its robustness properties. The CACSD, the DVP system, and, subject to availability, the experiments with the plant physical model and the real plant can be utilized for this purpose.

---

[12] Unless the model structure and the structure of the uncertainty set are explicitly defined

[13] The DVP model is richer and more comprehensive than the generated data-based model in compliance with the “linearized synthesis model – nonlinear evaluation model” concept [Gru2001]

3.1.6. Adequacy of the developed control system

In this section we will discuss only some general aspects related to the quality and adequacy of the results. The in-depth analysis and more concrete recommendations are provided in Chapter 4, where a specific class of plants is defined.

The ultimate outcome of the DVP-based development method is the control system. As it has been discussed in Section 2.3, a control system is adequate if the functional and nonfunctional requirements are met for ***both*** the DVP and the actual plant. The problem in this dissertation is that the control system is developed using some ***artificial*** data generated using the DVP rather than the real plant. The adequacy, therefore, can be proved only when the control system is tested with the ***actual plant***. However, even this is not enough as the ultimate adequacy test is the experience of the practical ***exploitation*** of the system.

Nevertheless, one can take some measures towards making the control system adequate. The first group of measures is related to insuring the adequacy and quality of all intermediate results (DVP, data-based model, robustness of the control system). The second group of measures concerns the comprehensive testing and validation by all available means including, for example, expert knowledge, information on the preceding generations of the plant and its control systems, existing independent models, physical models and prototypes, and, finally, the actual plant.

Let us consider the first group of measures related to the intermediate results. The quality and adequacy of the intermediate results have to be considered in the context of the whole development method keeping in mind the ultimate goal of obtaining the desired control system.

The quality of the data-based model generated with the DVP in the desired work regimes design may be assessed indirectly by employing the persistence of excitation and spectral characteristics of the input and output signals. The quality of the data-based model with respect to the applicability of the model for the subsequent control system improves if the persistent excitation order is high and the spectra are coherent with the frequency characteristic of the plant [Lju1999], see Section 4.2.1.1.

The model order selection criteria have to account for both the residuals and the model complexity [Lju1996, Ite2002, Ris2001]. The most commonly used model order selection criteria – the χ^2 criterion, Akaike's information criterion (AIC), and Rissanen's minimal description length criterion (MDL) – allow to select the optimum (with respect to a particular criterion) model order within a selected model class or, in case of MDL, across several model classes [Ris2001]. In practice, though, a fairly simple (e.g. low order, linear etc.) model of the plant in conjunction with a suitably robust control system may be sufficient [Ite2002].

One of the most straightforward ways of assessing the adequacy (dynamic similarity) of the identified model is validating the model on the fresh data set from the same dynamic regime, to which the identification data belongs [Lju1996]. The conditions of the identification and validation experiments may differ to some extent as the validation experiment aims at assessing the robustness of the model to the plant uncertainties or small variations of the dynamic regime (analogously to [Ite2002] where two types of experiments have to be conducted).

Since the models are used in the context of the control system design, it is reliable to compare models (e.g. the DVP and the identified analytical model) in a closed loop [Lan2001, LL1996].

If the parametric uncertainties are known and the synthesis of the locally robust controllers is possible, the gain scheduling method can tackle fairly large parameter variations [AW1995, Fun2001, Rob2002b].

The performance of the control system may be improved by fine tuning the parameters of the controller. Since the DVP allows to conduct multiple “trial-and-error” simulations of the control system, a very general approach is to formulate the control design problem as the optimization one with some suitable direct or indirect quality criteria[14] [Com1985, Gru2001].

3.1.7. Conditions of applicability

The *global* conditions of the DVP development method applicability include the following three groups of factors corresponding to the main stages of the method:

1. *Availability of the information* on the plant structure and parameters which are required for constructing an adequate plant DVP, as well as *the existence of the DVP system* capable of describing the physical phenomena of the plant relevant to the control loop;
2. *Possibility of generating a data-based model*, which adequately represents the dynamics of the plant and is suitable for the subsequent identification and control design;
3. *Availability of the mathematical methods* for identification and the control system design, which are applicable to the data-based dynamic model of the plant.

[14] This approach is quite laborious (computationally). However, scanning the parameter space is easier (cheaper, faster, safer etc.) with the DVP than with the real plant

3.2. Analysis of the method

3.2.1. Comparison with the existing development methods

The principal difference of the DVP-based control system development method from the conventional ones is in the ***organization*** of the development process (“***how***” vs. “what”). The proposed method can be positioned in between the “analytical” and the “physical” methods as it has the features of both of them, Figure 3.1.

The ***qualitative*** characteristics of the control system development methods are summarized in Table 3.1.

From the viewpoint of which aspect is the most significant for obtaining a model of the plant, the analytical modeling (representing the plant by means of formulae) is classified as a ***“knowledge-intensive”*** method, while the physical modeling (using a physical analog of the plant) or experimenting with the real plant are ***“data-intensive”*** ones [Hyo2001]. The knowledge-intensiveness means that the control system developer has to possess the knowledge and the skills of how to develop an analytical model of the plant. This process demands a large amount of the intellectual resources. The data-intensiveness means that there has to be a sufficient amount of real, physical data collected from real physical objects, which represent the plant.

By analogy, the DVP method can be categorized as ***“computationally-intensive”***. This underlines the nature of the differences between the three modeling methods and, therefore, the corresponding methods of the control system development. The DVP development method is ***complementary*** to the analytical and physical methods. It allows to reduce the knowledge-intensiveness of the modeling process (by automating it and integrating with the VP/CALS infrastructure) and to avoid expensive or infeasible physical experiments (by substituting the DVP for the physical model or the real plant).

This classification is not flat: each method requires the engineer to have knowledge of the plant and certain skills. For the “data-intensive” methods the engineer has to know how to implement physical experiments; the “knowledge-intensive” methods imply the skills in the mathematical representation of physical processes; for the “computationally-intensive” the proficiency in the DVP, CAD/CAM/CAE and CACSD technologies is essential.

The input information and the starting point for the analytical modeling and for obtaining the DVP are very similar. In both cases the information on the structure and parameters of the plant and the control problem is required. The difference between the two methods is in the further way of processing this information.

Table 3.1. Qualitative characteristics of the control system development methods

	Analytical modeling	**Dynamic virtual prototyping**	**Physical modeling or experimenting with the real plant**
Nature of the modeling process	Knowledge	Computing	Data
Input information	Mathematical model of the plant (structure and parameters)	Mathematical model of the plant (structure and parameters)	Physical model or a real plant
Mathematical description	Analytical (formulae); explicit	Analytical (formulae); implicit, encapsulated into the plant DVP	Not applicable
Representation of the model for the engineer	As the analytical model (formulae)	As the integral "physical" system	As the integral physical system
Relative quality of plant description	Deep	DVP – deep; single data-based model – shallow	Shallow (single data-based model)
Relative information capacity	High	DVP – high, single data-based model – low	High
Control system design	Model-based	Model-free & model-based with identification	Model-free & model-based with identification
Robust control system design	Analytical representation of the uncertainty set	It is possible to obtain only the generated data-based model; the uncertainty set is difficult to estimate	It is possible to obtain the real data-based model and to estimate the uncertainty set

Defining the dynamics of the plant in the explicit analytical form is not needed when the DVP systems are used. Constructing the DVP of the plant is a rather formal procedure of setting the structure and parameters of the plant, which resembles assembling the plant in accordance with the technical drawings. As the plant structure and parameters are set, the DVP system automatically constructs the equations of dynamics. The automatic construction of the equations is an important characteristic of the DVP systems. The necessary domain knowledge and the method[15] of constructing the equations are embedded into the DVP system only once by the developers of the DVP system, while the developers of DVPs exploit this functionality repeatedly. So, a portion of the intellectual costs is moved from the DVP developer to the DVP system developer.

However, it is the automation enabled by the information integration of the control system development process and the VP/CALS systems that is a novel, unique and significant feature of the proposed DVP method. This automation ***facilitates*** the information streams between the VP/CALS model and the control development and ***adapts*** a fairly abstract discipline of the control system engineering to the practical, industrial needs. These aspects will be analyzed in Section 3.2.2.

From the point of view of the computational and algorithmic efficiency, obtaining the data-based model of dynamics is simpler than constructing the analytical one. Especially as the analysis of the complex nonlinear plants, which have, for example, chaotic dynamics, in many cases has to be accomplished numerically rather than the analytically (see Section 4.1).

To sum up, constructing DVPs requires less intellectual and, therefore, financial and time resources than the analytical modeling.

The implicit analytical description of the plant dynamics is the basis of the DVP. Hence, the essence of the DVP is an analytical model. However, the form of the DVP is a physical object. The analytical description is encapsulated into the DVP and hidden from the control system developer. The DVP can be regarded as an alternative, somewhat intermediate, representation form of the plant along with the analytical and physical models and the plant itself.

The analytical model enables to gain a ***deep*** insight into the internal cause-and-effect relations of the plant, whereas the experiments with the physical model and the real plant reflect only ***shallow*** input-output relationships and are limited to the regimes, for which the

[15] Only is the method of constructing the equations, but not the equations themselves. This makes the DVP systems somewhat universally applicable in a certain domain. Thus, Lagrange's formalism is one of such methods for mechanical systems with known structure, parameters and finite kinematic graph (see Section 3.1.2)

experiments are conducted. In other words, the physical models represent the plant as a black or a grey box.

Despite the fact that constructing the DVP requires generally the same information on the structure and parameters of the plant and, therefore, in principle, the DVP contains the same information as the analytical model, a single data-based model[16] generated using the DVP reveals only some shallow knowledge about a particular experiment[17]. In conclusion, the DVP itself is capable of providing a deep[18] insight into the plant (the property of the analytical model), while a single data-based model can be treated as shallow (the property of the physical model).

On the whole, the information potential of the data-based models generated using the DVP is weaker than one of the physical model or the real plant. Indeed, the DVP is able to reproduce only those aspects of the plant which have been taken into account by the developer of the plant DVP, but the real plant contains the complete spectrum of phenomena including the aspects which may be unknown to the developer.

An advantage of the DVP is that it allows to obtain the data-based model, which is necessary for the control system design, ***without*** analytical modeling even if experimenting with the physical model or the real plant is ***impossible*** due to some financial, ecological or ethical reasons. In addition, unlike the physical objects, the DVP does not impose any limitations on the input signals while obtaining the data-based model. It makes it possible to simulate the plants, which are close to instability, and even some unstable plants.

In summary, the DVP is ***dual***[19]; it takes an intermediate position between the analytical and the physical models. Constructing the DVP and the data-based model requires less intellectual resources than constructing the analytical model. Also, experimenting with the DVP allows to obtain the data-based model of the plant if the physical model or the real plant is unavailable. The drawback of the DVP is that it is "informationally" weaker than both the analytical model (in case of a single data-based model) and the physical model.

---

[16] A model describing the plant in one regime (only local dynamic similarity is possible)

[17] The loss of information on the plant in the transition from the DVP to the data-based model could be avoided in some cases if, for example, the symbolic computations were used to obtain the analytical model within the DVP framework. If this approach were adopted, it would be computationally less efficient than the numeric simulation. Besides, it would not work in a number of real situations when it is impossible to create a comprehensive though practically useful analytical model

[18] A model describing a number of regimes (the limit dynamic similarity is possible) or all regimes (towards the total dynamic similarity)

[19] [NY2001] also points out to the duality of the virtual model ("the third generation models can be both phenomenological and deductive"), but this claim is not elaborated in [NY2001]

3.2.2. Exploitation of the DVP information potential

The proposed development method not only aims at automating the process of obtaining the plant dynamic model. One of the most important elements is that it exploits the information potential of the DVP as a system aggregation of a number of models which describe various aspects of the modeled plant.

Using the DVP as a system is an advantage of the proposed development method. It is only the model of dynamics that is directly involved in developing the control system. However, such a comprehensive and integral approach to the development by means of the DVP, which makes it possible to consider the control system design in the context of the entire lifecycle of the system, significantly improves the development and raises it to a qualitatively higher level.

The DVP-based control system development features the well-known benefits of the DVP, which are implied by the comprehensive, multi-faced approach to modeling all system-related aspects. Employing the DVP enables to [Bae2002, Rob2002a]:

- Mitigate risks and avoid late changes in the design;
- Reduce the production costs;
- Reduce the product development cycle and, therefore, shorten time-to-market;
- Improve quality by extensive modeling and simulation and minimize the number of errors and on-site adjustments;
- Provide means for the efficient reuse of solutions by introducing changes incrementally, first to the VP, and then to the actual system.

As it has already been mentioned, automation of constructing the equations of dynamics is not a unique feature of the DVP. In general, being an ***open*** solution, the proposed development method may be technically implemented by means of the bond graphs or MODELICA language in conjunction with symbolic computations [Gru2001, Bon2005, MCA2005, Dif2005]. Moreover, the DVPs may be less capable of modeling systems with several types of physical phenomena than the bond graphs or MODELICA.

But it is the ***automation due to the information integration*** among the multitude of virtual prototype components and sub-models that is an exclusive feature of the method proposed in this dissertation. In this dissertation, the information basis – the digital mock-up, the virtual prototype, the system aggregation of models – is shared by all stages of the system lifecycle. Whatever changes happen at any stage of the lifecycle, they are immediately reflected in the virtual prototype as if it were a real physical object. This synchronization process is in the very structure of the virtual prototype. It is automatic. It is the automatic updating of the virtual prototype and its sub-models in the concurrent engineering environment.

Another positive manifestation of this type of ***automation*** is that for a complex plant/system a significant amount of work has to be done to extract the necessary structural and parametric information on the plant from the underlying database of the virtual prototype. In this dissertation the problem is simplified because the control system development process is informationally integrated (not coupled!) with the virtual prototype[20]. With the proposed method, the engineer is able to obtain the model, which is required for the control system design, from the virtual prototype almost automatically. And the method of obtaining the model along with the implications of the method for the control system design process is described in this dissertation.

It is interesting that the DVP mitigates the fundamental contradictions of modeling. Consider the following two pairs of contradictions [Eff1989]:

- Completeness vs. simplicity of the model;
- Specialization vs. generality of the model.

The integrated DVP is complete (comprehensive) and universal (general) since the DVP unites a number of coherent specialized models. At the same time, a cross-section of the DVP, which is considered in this dissertation, can be regarded as a simplified and specialized model targeted at describing the dynamic properties of the plant. In essence, the DVP and its cross-sections are indivisible. Thus, on the one hand, the changes in the DVP are directly reflected in the model of dynamics and, on the other hand, the developed control system can be tested with the complete DVP.

3.2.3. Information integration of the control system development using DVP

An additional synergetic effect is reached by implementing the proposed development method within the (D)VP/CALS framework as it allows to integrate all stages of the system lifecycle including the control system development stage. When the common VP is used throughout the lifecycle, it improves the coordination of all participants of the lifecycle. The coordination and the possibility of assessing how certain modifications affect the individual stages of the lifecycle and the whole system are important because nowadays different departments and enterprises are responsible for different stages of the lifecycle or components

[20] Still, subject to the efficient methods for obtaining macro models

of the system. For example, the coordination makes it possible to eliminate the problems when one department changes the system (plant), but the control system developers operate with obsolete models and receive this updated information with a considerable delay.

Constructing the DVP is the most resource consuming stage of the proposed development method. According to [Wen2002], for some static virtual prototypes (in the domain of manufacturing) it takes four times longer to construct the DVP than to experiment with it. In this dissertation the situation is worse because the dynamics has to be taken into account, carefully modeled and validated[21].

Once the DVP is constructed, it makes sense to reuse it as much as possible gradually improving and updating it both within the current generation of the system and throughout the subsequent generations. This will help to accumulate the experience of development, specific algorithms, solutions and knowledge in the DVP.

A rational way is to construct the VP from the early stages of the system lifecycle, to enhance it from stage to stage and to use it throughout all stages including testing, manufacturing, personnel training [WML1998] and marketing. As a result, (1) the cost of constructing the DVP will be divided among the stages of the system lifecycle and (2) the control system developer will be able to concentrate on the designing the control system rather than on the constructing the DVP.

3.3. Efficiency and practical applicability of the method

The most significant effect of applying the proposed development method can be achieved when

- The analytical dynamic model of the plant is unavailable and the development of this model is resource consumptive, while experimenting with the physical model or the real plant is not possible (e.g. the plant may not physically exist, the plant is unstable, the experiments are inexpedient due to economic, ecological or other reasons);
- The analytical model is available, but it has to be validated using a model which is obtained with a relatively independent method. This is the case when the DVP method is employed in conjunction with the conventional ones (e.g. to validate the correctness of the analytical model obtained with paper and a pen).

[21] This conclusion complies with the experience of the author of this dissertation

- It is possible to reuse the DVP at all stages of system's lifecycle and to develop the DVP from generation to generation of the system in an evolutionary way. This decreases the relative costs of the DVP development.

The joint application of the models which are built with the analytical modeling and the DVP approach brings particularly fruitful results. It makes it possible to cross-check the models for both the subjective and objective defects. This conclusion has been confirmed by the author of this dissertation, while developing the analytical model and the DVPs of the test examples described in Chapter 5 (gantry crane) and in [KT2002, KT2003] (double-link pendulum).

As a conclusion, the proposed DVP-based development method complements the conventional methods rather than replaces them.

From the point of view of the control system developer, the proposed DVP-based method is conceptually novel. Unlike the classical control engineering, where it is necessary to describe the plant as a system of equations, the DVP represents the plant as a virtual "physical"-like object. The DVP offers the phenomenological approach[22] to representing the plant, which enables the control engineer to perceive the plant as an integral, holistic object or phenomenon. The phenomenological form of the DVP quite naturally implies the empirical data-based dynamic model, which is generated on the basis of the DVP.

The biological aspect of the DVP approach is that the phenomenological DVP facilitates the active ***concurrent*** operation of both the left (induction, deduction) and the right (analogy, association) brain hemispheres. This stimulates the heuristic and intuitive streams of the thought process[23] and facilitates the concurrent active operation of the brain hemispheres, the so-called spillover [Nad1990].

According to the experts [BDS2003, Bae2002, Rob2003, Wen2002], the ubiquitous application of the VP/CALS tools in industry appears to be inevitable and imminent[24]. The need for the qualitative improvement offered by the VP becomes more prominent in the globalizing world with the constantly increasing pace of development, complexity and quality of products.

---

[22] Phenomenological, i.e. dedicated "to the things themselves". Phenomenology is a philosophical stream oriented at "the careful ***description*** of phenomena in all areas of experience" [Phe1996]

[23] It is interesting that the slogan "see what you mean", intuition and heuristics have become the concept of the recent Dassault Systèmes promotion campaign (BBC World TV, October 2005)

[24] Including such countries as, for example, Russia [Inf2003, ZY2003, BDS2003]

As it has already been stated, the efficiency of the proposed DVP-based method increases when the DVP is used repeatedly at all stages of system's lifecycle. As a result, there is a strong dependence on how much the industry in a particular country relies on the CAD and VP/CALS tools. The VP/CALS systems are rather expensive. The cost of an average license of the VP/CALS system V5 of Dassault Systèmes may be as high as several tens of thousands of euros. There are additional costs of deployment, training, technical support, computing infrastructure and so on. Therefore, those companies which have already adopted the basic VP/CALS may be more willing to apply the proposed DVP-based control system development method within the existing infrastructure because (1) it will not require new substantial investments and (2) it will allow to make the most of the existing VP/CALS infrastructure.

While improving the quality of the development process is important for the users of the VP/CALS systems and the control engineers, the ***developers of the (D)VP/CALS*** systems may find the proposed DVP method as means to intensify the ***innovation*** component of their business and, as a result, to raise its competitiveness. "Instead of looking within the conventional boundaries that define how an industry competes, managers can look methodologically across them. By doing so, they can find unoccupied territory that represents real value innovation" [KM2001].

It is the DVP-based control system development that may become such "unoccupied territory" for the (D)VP/CALS developers. Indeed, the method, which is proposed in this dissertation, will not only enable further development of the traditional CAD and information technology component of the (D)VP/CALS systems, but it will also boost the component related to the fundamental extension of the (D)VP/CALS functionality to the control system development.

3.4. Summary

1. A new method of developing control systems on the basis of the DVP has been proposed and analyzed. The new method facilitates the control development process and expands the applicability of the DVP/CALS systems. The proposed method requires neither the explicit analytical model nor the physical experiments with the real plant. It has been shown that the method improves the information integration between the control system development and other stages of the system's lifecycle;

2. The adequacy and quality of the method results have been analyzed in the context that (1) the data-based model is DVP-generated (rather than obtained from the real plant) and (2) the information on the plant uncertainties is incomplete;

3. The comprehensive analysis of the dynamic virtual prototyping in the context of the dynamic modeling and control system development has been conducted. The duality of the DVP has been revealed. Thus, the DVP interior has the features of the analytical model, whereas the DVP exterior appears as the physical object. The system analysis, mathematical, control, philosophical, biological, and economic aspects of the DVP have been addressed;

4. The proposed DVP-based method has been compared with the conventional ones. It has been shown that the efficiency and quality of the control system development process can be improved by

- Automating the construction of the equations of dynamics (in the mathematical and the information integration terms);
- Integrating the control system development stage with other stages of the technical system lifecycle;
- Exploiting the DVP repeatedly at all stages of the lifecycle throughout several generations of the system;
- Performing the exhaustive and comprehensive simulation of the system represented by its integral, holistic, systemic DVP;

5. The applicability of the proposed method has been justified in the following cases:

- The analytical dynamic model of the plant is unavailable and difficult to obtain, while experimenting with the physical model or the real plant is not possible;
- The analytical model is available, but it has to be validated using a relatively independent model;
- It is possible to reuse the DVP at all stages of the system's lifecycle and to develop the DVP in an evolutionary way.

CHAPTER 4. ADAPTATION OF THE DVP CONTROL SYSTEM DEVELOPMENT METHOD TO A SELECTED CLASS OF PLANTS

This chapter is dedicated to adapting the proposed DVP-based control system development method to a specific class of plants. The main idea is as follows. If a DVP is adequately mapped onto some class (prototype space), then all formal mathematical methods applicable to this class can be also applied to the DVP mapping. It has been proved that using the DVP-based method it is possible to develop the desired control system for a plant, which belongs to the selected class. The nature of DVP imposes certain limitations on the conventional mathematical methods of identification and control. These limitations have been revealed and analyzed. In a way this chapter is designed to be a practical manual for the control engineer employing DVP as a new development tool.

This dissertation has a methodological character. It does not aim at proposing a new control design algorithm. On the contrary, the emphasis is placed on analyzing the ***joint*** application of the ***existing and well-known*** methods for modeling, identification and control in the context of the DVP development.

4.1. Definition of the plant class

The selected class includes deterministic and stochastic dynamic systems. The structure and parameters are assumed to be known. The observability and controllability of the plant is required. The functional operator of the plant has to be expressed by means of ordinary differential or difference equations. The plant can have insignificant (linearizable) nonlinearities and quasi-stationary parameters. The class incorporates all stable plants and those unstable plants which either can be stabilized by means of some analytical model-free method or remain within the phase space region, where the adequacy of the data-based model to the plant is required.

The selection of the above-described class is explained by the fact that for the plants that belong to this class (1) it is relatively easy to obtain the required data-based models and (2) there exist a number of reliable identification and control methods. Let us analyze the introduced class.

The necessary[1] condition for constructing a DVP of the plant is the availability of the information on the plant structure and parameters (including the plant inputs and outputs). The information has to correspond with the control problem and its system complexity.

The plant has to be completely observable and controllable [Zak2003]. This is a natural requirement of the identification and control methods when a data-based model is employed as a basis for the control design.

The functional operator of the plant usually belongs to the class of ordinary differential or difference equations in two cases. In the first case the plant has lumped parameters. In the second case the plant has distributed parameters, but an appropriate approximation for a lumped-parameter model can be made.

All real systems are to some extent nonlinear and nonstationary. Nevertheless, the existing well-developed formal identification and control methods, which are intended for linear systems, may be also applicable to the systems with linearizable nonlinearities and quasi-stationary parameters.

The basis of the proposed method is in generating the data-based models that describe the plant in the desired work regimes. While the DVP, as a rule, is nonlinear, the generated data-based model can be regarded as a linear approximation (not linearization) of the DVP in the corresponding work regimes if the variations of the phase variables are small. This makes it possible to employ the mathematical methods, which are intended for linear systems.

Finally, for linear systems there exist formal and constructive criteria and methods for insuring the quality of the data-based model.

The methods, which are intended for the single-regime stationary systems, may be extended to the multi-regime and quasi-stationary plants, for example, by means of the iterative identification and control methods [Ite2002, TS1999] (see Section 4.2.2.2) or the gain scheduling approach [AW1995, Fun2001, Sha1996, Sen2004, Koi2000]. Specifically, the gain scheduling method is applicable if

1. It is possible to cluster some dynamic regimes in the parameter space of the plant or its environment. For example, such regimes can be clustered in the “load mass – rope length” parameter space of the gantry crane, Figure 2.5;
2. A local control system can be designed for each of the regimes;

[1] Necessary, but not sufficient. There has to be some extra semantic information, which allows to assess the system complexity, the applicability of DVP and so on

3. The dynamics of the transitions between the regimes and, therefore, the dynamics of the gain scheduling process, is much slower than the dynamics of the local control systems to avoid the transient instability of the entire system.

The possibility of generating adequate data-based models depends on the stability of the open-loop plant. This issue has been discussed at length in Section 3.1.4. Should the available *a priori* information be insufficient, the direct simulation of the DVP has to be employed for assessing the plant stability. An asymptotically stable plant can be easily kept in the proximity of a stable stationary point, whereas for an unstable plant it is necessary to study the divergence rate of the phase trajectories. If the divergence is not fast, there may be a relatively simple way of stimulating the plant to return to the desired work region.

The fact that it is possible to develop the desired control system for a plant of the selected class has been proved in this chapter. The essence of the proof is demonstrating that there exist a coherent chain of modeling, identification and control system design methods, which lead to the desired solution of the control problem subject to the availability of the necessary information on the plant. It has been demonstrated that the coherent chain of methods does exist, but there are some specifics in using these methods in the context of the DVP method. In addition, it has been shown that the method is possible to be implemented by means of the contemporary DVP and CACSD systems.

In principle, the boundaries of the DVP-based development method can be extended on some significantly nonlinear systems. For example, nowadays chaotic systems attract much attention of the scientific community [AF2004, Chu2003]. The proposed DVP method can be applied to this class of systems. Thus, the “path following” method for analyzing complex dynamic systems and constructing their bifurcation diagrams [EKM2002, EK2003] as well as the method for computing the fractal dimensionality of a phase trajectory [Cat2000] can be conveniently implemented using DVP. Complex dynamic systems are sometimes so complicated that simulation becomes the only way to study the system behavior. The DVP method is useful because it automates the construction of the dynamic model. It helps the engineer to focus on the phenomenon rather than on constructing the analytical model, which anyway could be employed only on as a basis for simulation. In sum, analyzing chaotic and other nonlinear systems by means of the DVP is a very important future research direction.

The soft computing methods (fuzzy logic, neural nets, genetic algorithms) [HMB1993, Fuz1999, Zak2003, Koi2000] are also capable of extending the applicability of the DVP method. Conceptually, the main advantage of the soft computing is that it allows to tackle nonlinear multi-regime systems while using data-based models.

4.2. Issues of the identification and control design methods in the context of DVP

In this section we will analyze the specifics of applying the existing, conventional, mathematical methods of identification and control system design in the context of DVP. The following two groups of methods are considered.

The control system design methods of the first group require an analytical model of the plant to be identified from a generated data-based model. The second group of methods uses the generated data-based model directly.

The boundaries of the two groups are not very strict. On the one hand, both groups explicitly or implicitly involve identification. On the other hand, the first group relies on the analytical model, which is more formal, abstracted from the real plant than the data-based model. In contrast, the second group does not assume that a "true" analytical model exists.

Virtually any method of identification and control can be employed in the DVP framework. However, using the DVP instead of the real plant implies some difficulties, which are primarily due to the insufficiency of information on the model uncertainties.

We will consider only those methods of identification and control which satisfy the criteria formulated below. A method has to be

- Mathematically formal;
- Applicable to multi-input multi-output (MIMO) plants;
- Well-developed from the engineering viewpoint;
- Realizable in the DVP and CACSD systems.

4.2.1. Analytical model identification and control system design

The methods of this group contain the following two steps: the analytical model identification and the control system design based on the identified analytical model. We have to bear in mind that identification is only an intermediate step and the main quality criterion of identification has to be directly associated with the control system. Also, being closely related to the real plant, the identification methods demand a more creative approach than the control design methods, which are abstracted from the plant by the identified model.

The (parametric) identification process comprises three main steps [Dei1979]:

- Selecting the structure of the model;
- Selecting the performance criterion;
- Determining the optimal model parameters with respect to the performance criterion.

The methods of identification fall into several groups [Lju1999, Dei1979]:

- Frequency domain (transfer matrix model) vs. time domain (state-space model);
- Active (an active input signal is used) vs. passive (the normal operation of the plant is not disturbed);
- Open-loop vs. closed-loop (with respect to the control loop);
- Iterative (e.g. recursive) vs. non-iterative (batch).

The most formal control design methods for MIMO plants, for example, the linear-quadratic (LQ) control, operate with the state-space linear models. Moreover, there are quite formal methods for identifying the parameters of the state-space linear models, for instance, the subspace identification method [VD1996, Lju1999], which is discussed in detail in Section 4.2.2.1.

An overview of the classical frequency domain identification of transfer matrices, which are involved in Section 4.2.1.2, can be found in [Lju1999, Lju1996, Dei1979].

4.2.1.1. Identification of the analytical model

There exist formal and constructive criteria and methods of insuring the quality of the identified linear systems.

The quality of data (data-based model) means that they are suitable either for the identification of an analytical model or for the direct data-based control system design. The following three factors define the information capacity of the data-based [Lju1999, AW1995]: the persistence of excitation order, the number of samples, and the spectrum of the input and output. Moreover, the positive role of the closed loop identification has to be taken into account [Lan2001].

Formally, a linear system is identifiable if it is completely observable and controllable and the input signal is *persistently exciting* [Lju1999].

Persistence of excitation is a quantitative measure of whether the input signal is “sufficiently changing”. An input signal is persistently exciting if the input covariance matrix is invertible. The matrix is invertible when the *persistence of excitation order* of the signal is not less than the size of the full rank covariance matrix.

This concept can be illustrated using the classical least squares method [AW1995]. Consider the following scalar regression model:

$$y(t) = \sum_{i=1}^{n} \theta_i x(t-i) + \varepsilon(t), \tag{4.1}$$

where t is the discrete time, $x(t)$ is the input, $y(t)$ is the output, $\varepsilon(t)$ is the additive zero mean normally distributed noise, θ is the model parameter vector, n is the model order. An estimate of the parameter vector $\hat{\theta}$ in (4.1) is given by the least squares solution:

$$\hat{\theta} = \left(X^T X\right)^{-1} X^T Y \,, \tag{4.2}$$

where $Y \in \Re^{k\times 1}$ is the output (measurement) vector, $X \in \Re^{k\times n}$ is the input matrix, $X^T X$ is the input "covariance" matrix[2], k is the number of measurements. The input matrix X has the following structure:

$$X = \begin{pmatrix} x(t-1) & x(t-2) & \dots & x(t-n) \\ x(t) & x(t-1) & \dots & x(t-n+1) \\ \dots & \dots & \dots & \dots \\ x(t+k-2) & x(t+k-3) & \dots & x(t+k-n-1) \end{pmatrix}.$$

The least squares solution (4.2) exists if the input covariance matrix $X^T X$ is invertible, i.e. has full rang. The persistence of excitation order in this case should be greater than n. It is easy to show that if the signal does not change enough (there are repeated blocks in the signal $x(t)$), the rows of X will be almost linearly dependent, as well as the rows of $X^T X$, leading to the difficulties in inverting $X^T X$ and, therefore, obtaining a meaningful solution.

In the case of the subspace identification method [VD1996], the size of the input covariance matrix is: $2mi$, where m is the dimensionality of the input signal, i is a parameter of the method with $i >> n$, n is the order of the identified system. For a solution of the identification problem to exist, the input signal must be such that the rank of the input covariance matrix is $2mi$. In other words, the persistent excitation order of the input vector signal must exceed $2i$.

The orders of the persistence of excitation for some standard signals are given in [AW1995]:

- Step function 1(t): *1*;
- Sinusoid: *2*;
- Signal containing k different harmonics in its spectrum: *2k*;
- Periodic signal of period *N*: ***not more than*** *N*;
- Symmetric square wave (meander): *N/2*;
- White noise[3]: ∞.

---

[2] In fact, for zero-mean data the sample covariance is $\dfrac{1}{n-1} X^T X$

[3] An uncorrelated sequence of samples is sufficient

In practice, there are more requirements to the input signal. Thus, the quality of identification depends on the *energy distribution in the spectrum of the input signal* [Lju1999, LL1996]. Indeed, the identified model has to accurately describe the plant (the DVP) in the bandwidth of the closed loop system. In many cases the bandwidth of the closed loop system is located in higher frequencies than the bandwidth of the open loop system (e.g. when the purpose of the control system is to reduce the transient time).

On the one hand, the spectrum of the input signal has to be concentrated[4] in the high frequencies to obtain an accurate description of the closed-loop system (the input spectrum is limited from below). On the other hand, the input spectrum has to cover a large part of the open-loop system bandwidth (the input spectrum is limited from above). Otherwise, if a plant is a low-pass filter, a high frequency input signal will be considerably subdued by the plant leading to a “loss” of connection between the input and the output. Additionally, the spectrum of the input signal should cover possible resonant frequencies of the plant.

In conclusion, experimenting with the input spectrum is likely to improve the quality of the identified model.

Furthermore, the identification results depend on the distribution of the input energy in time [Lju1999], homogeneity of the data-based model (each model has to cover only one dynamic regime), and the proper balancing (scaling) of the plant [SP1996].

The length of the data-based model (the number of samples) is algorithm-dependent. The lower bound is imposed by the specific algorithm requirements (insuring the rank of the input covariance matrix, decreasing the variance of the estimated model parameters) and the time scale of the plant dynamics, whereas the upper bound is due to the maximum allowed computational complexity (or duration) of the DVP experiment.

Note that not only do the described factors, i.e. persistence of excitation, number of samples, spectral characteristics, define the quality of the data-based model, but they also indicate the directions of how the data-based model can be improved. An advantage of the DVP method is that it allows to generate even rather complex-shaped high-energy input signals.

The *subspace identification* is one of the most convenient modern methods for identifying linear models in the state-space [VD1996, Lju1999]. It can be classified as a parametric time-domain active (open-loop) non-iterative identification method.

---

[4] Some concentration is essential as all real signals carry a limited amount of energy

The main idea of the method [Lju1995, Lju1999] is that the state vector of the plant (up to a similarity transformation of the vector space basis) can be reconstructed using only the input and the output data without any prior knowledge of the *(A,B,C,D)* matrices. Specifically, the $\overline{1,n}$-step-ahead predictors, which use the retrospective input/output data, are employed. As soon as the state vector is obtained, the problem of fining the *(A,B,C,D)* matrices reduces to the conventional least squares method.

There are deterministic, stochastic (the plant input and output are stochastic processes) and combined subspace modifications of the method. The recursive version of the subspace method is presented in [LGV1998]. Subspace identification from closed-loop data is discussed in [Ver1993, LM1996, KKP2005].

The open-loop variant of the method is implemented in the `n4sid` function of MATLAB / Identification Toolbox [MLAB2005] and the `findABCD` function of Scilab / Identification [SLAB2005]. The general requirements to the data-based models do apply to the subspace algorithm.

The subspace method has been extensively studied in the literature in the recent years. The basic error analysis of the subspace estimates of the system matrices was addressed in detail in [CP2004a, CP2004b]. The poor results of the subspace method are found to be related to ill-conditioning of the underlying regression problem, which happens in two cases. First, the collinearity of the state and future input subspaces. Second, the lack of excitation of the input signal. The asymptotic normality of the estimates (under some restrictions) is shown in [BJ2000]. The asymptotic behavior of the subspace method is also studied in [Pin2002].

Despite the strict analytical estimates for the variance of the identified parameters being not yet available for the subspace method, practically the condition of the problem can be assessed by means of the singular value decomposition performed by `n4sid`. If the condition is poor, it is an indicator that the quality of the input signal is insufficient.

When the sampling period tends to zero ($T \to 0$), it may be very difficult to obtain useful estimates of the model parameters [MG1990] because

$$A_d = e^{A_c T} \xrightarrow{T \to 0} I,\ B_d = \int_0^T e^{A_c t} B_c dt \xrightarrow{T \to 0} B_c T\,,$$

where A_c, B_c are the matrices of the continuous system; A_d, B_d are the matrices of the discrete one [Bur2001]. Numerically it is better to work with small quantities (the operands are close to zero) than with small differences of large quantities (the operands are close to one, but it is their difference that is of importance).

Hence, it may be better to identify the matrix $A_d–I$ instead of the matrix A_d, if the sampling period cannot be increased. This approach is called the δ-operator[5]. The subspace identification method in conjunction with the δ-operator has been studied in [HK2000, HK2001].

In this dissertation the subspace method has been successfully applied to identifying the analytical model of a gantry crane (see Chapter 5). The simulation experiments indicated that the quality of the subspace identification improves as the stability degree of a plant increases and the variability of ***both*** the input and the output signals grows[6].

In order to *estimate an appropriate model order*, there are some constructive criteria accounting for both the residuals of the model and its complexity, which, as a rule, increases as the model order raises. The most widely used criteria are H. Akaike's information criterion [Aka1983, Sod2002, Arn2004, Gus2000] and J. Rissanen's minimum description length criterion (MDL) [Ris1978, Ris2001, Gus2000]. For normally distributed disturbances the criteria have the following form: [Lju1995]:

$$\begin{aligned} AIC &= \left(1+\frac{2\dim\theta}{N}\right)\frac{1}{N}\sum_{t=1}^{N}\varepsilon^2(t,\theta); \\ MDL &= \left(1+\frac{\log N\,\dim\theta}{N}\right)\frac{1}{N}\sum_{t=1}^{N}\varepsilon^2(t,\theta) \end{aligned},$$

where N is the length of the data-based model, $dim\ \theta$ is the number of analytical model estimated parameters, and $\varepsilon^2(t,\theta)$ are the squared residuals.

Unlike AIC[7], theoretically the MDL criterion enables to compare models, which belong to different classes. The essence of the MDL principle is as follows [Ris2001]. Suppose there are some digital data to be transmitted as a signal over a communication channel. The signal has to be compressed to reduce the amount of transmitted information. To

---

[5] In the literature there is not enough attention to the relations between the quality of the subspace estimates and the dynamics of the plant. Apparently, there are indications that the problems related to small sampling periods ($T\to 0 \Rightarrow A\to I$), poor stability of the plant ($|eig(A)|\to 1$), and the previously described loss of connection between input and output for low-frequency plants ($eig(A)\to I$) are the manifestations of the same ill-conditioning problem when the variability of the state vector sequence almost entirely depends on the variability of the input signal (regardless of the input persistency of excitation). These phenomena have to be further investigated

[6] The variability was achieved by increasing the persistence of excitation and the energy of the input signal. At the same time, the variability of the output signal was achieved by insuring that the input spectrum covers the low frequencies, where the bandwidth of the open loop gantry crane is located

[7] In practice, both MDL and AIC usually lead to the same conclusion for models belonging to the same class [Gus2000]

compress the signal it is necessary to find a simple model, which would accurately describe the signal, and to transmit the parameters of the model along with the residuals between the compressed and uncompressed signals. The better the model is, the better it describes the signal and the smaller the residuals are. Conversely, there are many parameters to transmit if the model becomes very accurate. The problem is to find such a model that the overall amount of transmitted information is minimized.

Apart from AIC and MDL, the classical methods of the singular analysis [KNM1989], correlation analysis and statistical testing (e.g. the χ^2 method for testing the residuals distribution) [Sod2002, Pug1979] can be employed for estimating the model order.

An identified model has to be *validated*, i.e. compared with the DVP or some analytical model (if it is available). The validation can be accomplished with some data, which are different from the identification data, but belong to the same dynamic regime [Lju1999, Ite2002] [8]. While the identification data (the data-based model) has to be suitable for identification (persistence of excitation, number of samples, spectrum), the purpose of the validation data is different as it has to show that the identified model is indeed similar to the DVP in the dynamic regimes in question. Here we also note that the squared-residuals-based distance measures are very sensitive to outliers [Ite2002, Hyo2001]. The robust methods, which use absolute values or even signs of the residuals, are preferable.

Comparing models *in a closed loop* is known to produce reliable results [Ite2002, Lan2001, LL1996]. If two systems (models) have similar behavior in an open loop, it does not imply that the two systems will similarly behave in a closed loop [Ite2002]. Comparing two systems in an open loop does not make sense at all if at least one of them is unstable or marginally stable. As a result, models have to be compared in a closed loop employing such (proximity) criteria, for instance, as the residuals norm, the closeness of poles or step responses.

Moreover, in [Ite2002, Lan2001, LL1996] it is also recommended to re-design the control system through the data-based model obtained in a closed loop (with the initial control system). This may improve the quality of the control system because the analytical model, which is identified with the closed loop data-based model, better describes the plant in the ***actual*** closed loop operation. The algorithm is as follows [LL1996]:

[8] The cross-validation method can be used if there are not enough data. Though, the amount of data should not be a problem with the DVP method as it is relatively easy to generate a lot of data. A drawback of the cross-validation method is that the information capacity of the data-based model is reduced. Besides, it is necessary to keep in mind the Nyquist-Shannon-Kotelnikov conditions

1. Identify an analytical model of a plant with its open-loop data-based model;
2. Design an initial control system for this analytical model;
3. Simulate the control system in a closed loop and obtain the closed-loop data-based model;
4. Identify a new analytical model with the closed-loop data-based model[9] and re-design the control system.

4.2.1.2. Control system design on the basis of the analytical model

The design of linear control systems on the basis of the plant analytical model is a subject of the classical automatic control theory. Tens of methods have been developed for this purpose. From the practical point of view one has to mention the *proportional-integral-derivative* (PID) control, which is found in a large number of industrial systems because of its simplicity, reliability and robustness [AW1995, PID2001], and the *linear-quadratic* (LQ) or *linear-quadratic Gaussian* (LQG) control due to its high level of formalism and applicability to control *multi-input multi-output* (MIMO) plants.

Employing the PID controller in the DVP development context does not have any specifics in comparison with the classical application. On the contrary, there are certain difficulties with the LQ (LQG) controllers.

The LQ is known to have very strong ***robustness*** properties with respect to the ***nonparametric*** uncertainties in the control loop at the input of the plant subject to the controller being designed for an accurate plant model [SA1997]. The multivariable phase margin of the LQ is ±60° and the gain margin is infinite upwards and 0.5 downwards. However, this result is extremely ***sensitive*** to the ***parametric*** uncertainties of the matrices *(A,B,C,D)* [LA1996]. In the DVP context, it is ***critical*** since the identification with the data-based model results in some approximation of these matrices. Moreover, the data-based model is generated by the DVP rather than obtained from the real plant. This fact brings even more uncertainty into the *(A,B,C,D)* matrices.

The DVP development framework makes it almost impossible to use such improvements of the LQ as the sensitivity weights [LA1996] to tackle the ***known***[10] ***parametric*** uncertainties. The sensitivity weights method involves calculating the partial derivatives of the matrices A and B with respect to the uncertain parameters. In principle, if the variations are known, the derivatives could be calculated numerically, but the results

[9] All necessary measures to avoid biased identification results have to be taken [Lju1999]
[10] Those uncertainties that can be explicitly accounted for in the DVP

would be unreliable. The matrices A and B are ***identified*** (i.e. estimated) with the data-based models generated by the DVP and it is the inaccuracies of the identification that will be emphasized by the numerical differentiation.

Furthermore, the above-mentioned robustness properties of the LQ controller do not hold for the LQG controller, i.e. when the state vector is incompletely measurable and has to be estimated.

The fundamental difficulty of the DVP-based control system development is the insufficiency of the information on the model uncertainty set. In fact, there are two difficulties:

1. It is not known how exactly an uncertainty enters the model;
2. The DVP may not reflect the actual (real) uncertainties of the plant.

The first difficulty is common to all data-based control system design methods, whereas the second one is a feature of the analytical-model-based methods. Not only does the DVP method accumulate the advantages of the conventional methods (analytical and physical), but it also accumulates their difficulties.

It is impossible to avoid the second difficulty without an explicit reference to the real plant. The first difficulty could be circumvented mathematically if there were appropriate formal MIMO control system design methods, which do not require the explicit knowledge of the uncertainty set structure. However, such ***formal*** design methods are not readily available, especially for the state-space models.

But the construction of nominal models and uncertainty sets from the frequency-domain data has been studied quite extensively in the recent years, for example, in [BHN2004, RGL2002, HSB2002, KL2001]. Let us formulate two robustness-related problems, which can be solved using the DVP method ***assuming*** that the DVP is an adequate ("true") model of the plant. Both of these problems and the approaches to solving them have been inspired by [Ite2002] and [Rob2002b]. In fact, the nonparametric uncertainty problem can be solved directly by the method proposed in [Ite2002]. Yet, a different approach can be taken for the parametric uncertainties problem.

<u>Problem 1:</u> Parametric uncertainties. If multiple simulations of the plant DVP in the parameter space are performed, it may be possible to find out how the varying parameters affect the identified analytical model.

Suppose a MIMO plant, which has m inputs and n outputs, is described in the frequency domain by the following transfer matrix:

$$G(j\omega) = (g_{ik}(j\omega)) = \begin{pmatrix} g_{11}(j\omega) & \dots & g_{1m}(j\omega) \\ \dots & \dots & \dots \\ g_{n1}(j\omega) & \dots & g_{nm}(j\omega) \end{pmatrix}. \quad (4.3)$$

Let $p^{(0)}$ be the nominal set of parameters, and $p^{(l)}$ a particular combination of the varied parameters in the parameters space $P: p^{(0)}, \forall l \; p^{(l)} \in P$. The nominal model can be represented as $G_0(j\omega) = \left(g_{ik}^{(0)}(j\omega)\right) = \left(g_{ik}\left(j\omega, p^{(0)}\right)\right)$, and a perturbed model can be written as: $G_l(j\omega) = \left(g_{ik}^{(l)}(j\omega)\right) = \left(g_{ik}\left(j\omega, p^{(l)}\right)\right)$.

Step 1: Identifying the nominal open-loop model $G_0(j\omega)$. Set the nominal parameters $p^{(0)}$, generate the data-based frequency domain model, and identify the nominal analytical model. The data-based frequency domain model can be generated in the DVP system as follows. Define a frequency grid (most probably logarithmic). The grid has to expand at least up to the closed-loop crossover frequency. Apply the sinusoidal waves according to the grid to each input of the plant individually (one-by-one). Each sinusoid has the frequency from the grid and some reference amplitude. Measure the amplitude of the sinusoids at the outputs of the plant DVP (for linear plants the frequencies of the input and output sinusoids are known to be the same). If the grid has M nodes, there have to be M simulations for each of m inputs with total $M \times m \times n$ output measurements. Each *1*-input-to-*n*-output experiment yields the column in (4.3), which corresponds to the active input[11].

Step 2: Defining a grid in the parameter space. This is illustrated in Figure 4.1 for the two-dimensional case $p^{(l)} \in \Re^2$.

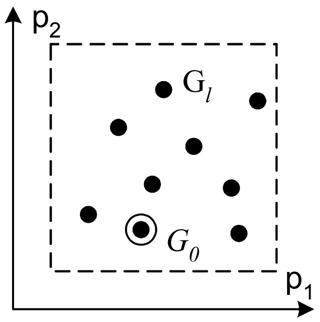

Figure 4.1. Parameter space and a grid in the two-dimensional case

[11] An alternative approach has been used in [KBTD2000]. Instead of using M individual sinusoids, a pseudo-noise signal of bandwidth $[0 \ldots \omega_{closed\ loop\ crossover}]$ is employed to excite a number of the plant modes simultaneously. The time-domain data is transformed to the frequency-domain via fast Fourier transform. The selection of the input signals is a crucial question because the transfer matrix may be ill-conditioned with large variations between individual $g_{ik}(j\omega)$. The presence of resonant peaks also imposes some restrictions on the minimum density of the frequency grid

Step 3: Performing multiple simulations of the open-loop plant DVP on this grid. For each node of the grid set the corresponding varied parameters $p^{(l)}$, and generate the data-based frequency domain models:

$$G_l(j\omega), \quad l=\overline{1,N}, \quad \omega \in \left[0 \ldots \omega_{\substack{closed\ loop\ crossover\\ frequency}}\right],$$

where N is the number of the grid nodes. It is very important that we do ***not*** have to identify the transfer matrix $G_l(j\omega)$ as only the empirical (data-based) frequency characteristic for each node is needed.

Step 4: Constructing the standard model [DFT1990, Toi2000]: $G = G_0 + W_1 \Delta W_2$, where $W_1 \Delta$ is the additive uncertainty, ΔW_2 is the multiplicative uncertainty, $\|\Delta\|_\infty \le 1$. Let us consider only the additive uncertainties. Then, the standard model can be written as:

$$G(j\omega) = G_0(j\omega) + W_A(j\omega)\Delta(j\omega), \|\Delta\|_\infty \le 1, \tag{4.4}$$

where $W_A(j\omega)$ is the additive uncertainty weighting transfer function (SISO), which can be defined using the following formula:

$$\begin{aligned} &|W_A(j\omega)\Delta_{ik}(j\omega)| \ge \left|g_{ik}^{(0)}(j\omega) - g_{ik}^{(l)}(j\omega)\right| \Big/ \|\Delta(j\omega)\|_\infty \le 1 \\ &\forall\, \omega \in \left[0 \ldots \omega_{\substack{closed\ loop\ crossover\\ frequency}}\right],\ l=\overline{1,N} \\ &i=\overline{1,n};\ k=\overline{1,m} \end{aligned} \tag{4.5}$$

The meaning of (4.5) is that the scaled $W_A(j\omega)$ must envelope[12] (as tightly as possible) all element-wise differences $\left|g_{ik}^{(0)}(j\omega) - g_{ik}^{(l)}(j\omega)\right|$ between the nominal model and each of l perturbed models[13]. The scaling transfer function $\Delta_{ik}(j\omega)$ is arbitrary subject to $\|\Delta(j\omega)\|_\infty \le 1$ (the H_∞-norm of the whole matrix should not exceed one). The element-wise differences can be calculated frequency-by-frequency using the data-based frequency domain transfer functions obtained at Step 3.

Problem 2: Nonparametric uncertainties. The sources of the nonparametric uncertainties are some unaccounted, neglected dynamic effects. In the context of this

[12] With respect to the amplitudes; the phases do not matter due to the norm-bounded uncertainty

[13] From the differences $\left|g_{ik}^{(0)}(j\omega) - g_{ik}^{(l)}(j\omega)\right|$ in the numerical form, $W_A(j\omega)$ in the analytical form must be found

dissertation it means that there are some differences between the plant DVP and the linearized nominal model due to, for example, linearization effects and identification inaccuracies. The standard form (4.4) can be obtained very easily. Instead of *l* experiments we need just one. First, the nominal analytical model is identified based on this experiment. Second, the difference between the analytical model and the data-based model of the experiment is calculated. The Problem 2 is virtually incorporated into the Problem 1.

As soon as the standard form (4.4) is constructed, we can solve a number of problems. First, instead of designing a control system for the nominal plant we can design a robust control system for the uncertain system (4.4), which describes the entire parameter space, using the conventional robust control methods [DFT1990, SP1996]. Second, since the experiments are conducted in an open loop, we can, for example, test the stability of ***a number*** of controllers in the parameter space with no need to conduct multiple simulations for each tested controller.

The procedure for testing whether controller *K* is robustly stabile is as follows [Toi2000]. Calculate the H_∞-norm for the generalized plant: $F_{v\to z}(G,K)=(I-KG_0)^{-1}KW_A$, Figure 4.1. If the nominal plant is stable and $\|F(G_0,K)\|_\infty<1$, then the control system with the controller *K* is robustly stable for any bounded-norm uncertainty $\|\Delta\|_\infty\le 1$.

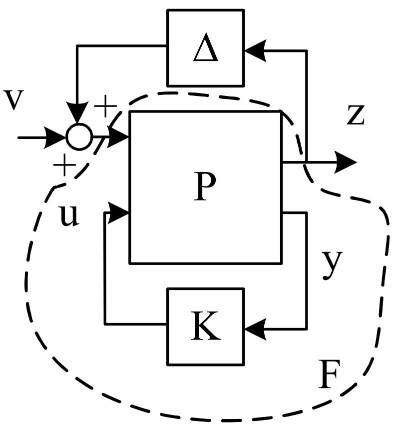

Figure 4.2. Standard representation of the uncertain control system $y=(G_0+W_A\Delta)u$ *; K is a controller; P is an augmented plant defined by* $z=u$, $y=W_Av+G_0u$ *; u – control input, y – measured output, z – controlled output, v – disturbance [Toi2000]*

The suggested straightforward procedures are rather immature. Moreover, similar, although more advanced, solutions have been proposed in [BHN2004]. Here we have provided only a rough ***sketch*** of the robustness-related problems and the approaches to solving them. Thus, we have not defined how to set the parameter grid (its structure and the number of nodes), how to set the frequency grids (the frequency range and the number of nodes), how reliable the constructed general form (4.4) is. Furthermore, if the dimensionality of the parameter space increases and dense grids are used, the computational complexity of the suggested procedure of constructing (4.4) will be enormous. Nevertheless, this direction appears to be very significant and it deserves more attention in the future research.

The uncertain frequency-domain models discussed in this section are continuous-time. It is convenient because the classical literature on robust control mostly refers to the continuous-time analysis and design methods. A digital controller can be obtained from the continuous-time one by sampling. However, sampling does not preserve the robustness properties (e.g. can be ruined by an improper sampling period) [Rob2002b]. Therefore, the robustness of the digital controller has to be re-verified.

In [Rob2002b] the following two methods are suggested for the analysis of the parametric robustness: Mikhailov's uncertainty sets and Kharitonov's interval polynomials, which are suitable for a large number of uncertain parameters when the direct multiple simulations in the parameter space would be inefficient. But neither of these methods works for an arbitrary plant in the DVP framework. In Mikhailov's method, the uncertain parameters must enter the coefficients of the characteristic polynomial linearly and only one parameter per coefficient. This cannot be guaranteed for an arbitrary plant since the DVP approach uses data-based models. In Kharitonov's method, it is necessary to know the range of the uncertain coefficients of the characteristic polynomial. In addition, Kharitonov's method is only sufficient (i.e. may lead to over-conservative results) if the variation of the coefficients is not truly independent. Neither requirement can be satisfied without multiple simulation experiments in the parameter space.

Therefore, the simulation experiments on a grid in the parameter space seem to be the only choice for the analysis of the robust stability and performance in the DVP framework. However, this approach may be computationally difficult, especially for dense grids and a large number of uncertain parameters. Another obstacle with checking the stability of the control system by simulation is that the length of the simulation experiment is finite and there is no guarantee that the system does not diverge as $t \to \infty$.

To sum up, a bottleneck of the proposed DVP-based method is the insufficiency of the information on the model uncertainty set and the lack of the formal MIMO control system design methods, which would not require the explicit knowledge of the uncertainty set structure. Therefore, this makes the thorough and comprehensive simulations of the designed control system important.

4.2.2. Data-based control system design

The methods of this group directly exploit the data-based model of the plant and do not involve an intermediate identification step. The aspects regarding the quality of the data-based model and the relations between identification and the control system design, which have been discussed in the previous sections, also apply to the data-based methods. This is because the data-based methods ***implicitly*** employ identification.

The data-based methods have had a long history. The first methods were proposed by J. Ziegler and N. Nichols in the 1940's [ZN1942]. These methods make it possible to tune the parameters of a PID controller using the empirical (i.e. data-based) frequency or transient characteristics of the plant. The rules for tuning a digital PID controller were developed by Y. Takahashi in the 1970's [TCA1971][14]. The methods for the automatic tuning of the PID controllers belong to the same group [HA1996].

Nowadays the data-based methods are gaining an increasing attention of the control engineering community. These methods are especially useful when it is difficult to obtain an analytical model of the plant, but there are a lot of data describing the closed or the open loop plant [Ite2002].

The ***synergy*** of the data-based methods and the DVP control system development is that the necessary data-based method can be generated in a DVP system without the need to have a physical model or a real plant.

In this group we will consider the data-based linear-quadratic Gaussian (LQG) and the iterative identification and control method. These methods have been developed and considerably matured in the course of the past decade.

4.2.2.1. Data-based LQG control

The data-based linear-quadratic Gaussian (LQG) method is very well formalized and combines the subspace identification and the linear-quadratic (LQ) control [FDVG1998a]. The conventional LQG control minimizes the H_2-norm[15] of a closed loop system. This is equivalent to the combination of Kalman's observer and the LQ controller [LGA1996].

[14] The author of this dissertation has successfully applied Takahashi's rules in the DVP context for controlling a double-link pendulum

[15] H_2-norm equals the output energy of a system for the δ-function input; H_2-norm equals the mean output energy of a system for the white noise input [Toi2000, LGA1996]

Unlike the conventional LQ, designing a data-based LQG controller does not require the analytical model of the plant.

The problem formulation is given in [FDVG1998a]. Using the retrospective data of the input u_k and output y_k of the plant for $k<0$ find such an input sequence $(u_1, u_2, \ldots, u_N)$, which minimizes the following quality functional over the time horizon of the length N

$$J = \sum_{k=1}^{N} \left\{ (\hat{y}_k - r_k)^T Q_k (\hat{y}_k - r_k) + u_k{}^T R_k u_k \right\} \tag{4.6}$$

where $\hat{y}_k \in \Re^{m \times 1}$ is an estimate of the future output of the plant for k steps ahead, $u_k \in \Re^{l \times 1}$ is the control sequence for $k>0$; $r_k \in \Re^{m \times 1}$ is the reference signal (the desired output). The weighting non-negative definite matrix $Q_k \in \Re^{m \times m}$ and positive definite[16] matrix $R_k \in \Re^{l \times l}$ (l is the plant input dimension; m is the plant output dimension) are the design parameters defined by the engineer[17].

This problem formulation differs from the conventional LQG in two ways. First, (4.6) has a ***finite*** optimization horizon N. Second, instead of the ***state*** variables, (4.6) contains the discrepancy between the actual ***output*** and the reference signal (i.e. reference tracking problem).

According to the algorithm [FDVG1998a], two stationary matrices have to be calculated ***off-line***. The first one, $L_w \in \Re^{Nm \times N(m+l)}$, has a connection to the Kalman filter, whereas the second one, $L_u \in \Re^{Nm \times Nl}$, corresponds to the feedback control. The control law has the form of a stationary (with constant parameters) negative feedback calculated ***on-line*** at each step with updated r_f и w_p by the following formulae:

$$\begin{aligned} u_{k>0} &= \left(R + L_u{}^T Q L_u\right)^{-1} L_u{}^T Q \left(r_f - L_w w_p\right) \\ w_p &= \left(y_{k-N+1}^T, \ldots, y_{k-1}^T, y_k^T \mid u_{k-N+1}^T, \ldots, u_{k-1}^T, u_k^T\right)^T \end{aligned} \tag{4.7}$$

where $u_f = \left(u_1^T, \;\; u_2^T, \;\; \ldots \;\; u_N^T\right)^T \in \Re^{Nl \times 1}$ is the sequence of the future input vectors up to step N, $w_p \in \Re^{N(m+l) \times 1}$ is the auxiliary vector composed of the past inputs and outputs, $r_f \in \Re^{Nm \times 1}$ is the reference for N steps ahead, $Q = diag(Q_1, \ldots, Q_N) \in \Re^{Nm \times Nm}$ and

---

[16] Apparently, there is a mistake in the original article [FDGV1998a], which defines the Q and R matrices as non-negative definite. In this case R may be zero. However, this will contradict the formulated control problem. The quality functional J (4.6) is supposed to be minimized by means of an appropriate control sequence u. If R is zero, J will not depend on u yielding an "un-controllable" problem (this mistake has been found by Professor R. Stroganov)

[17] In the form (4.6) the data-based LQG lack the integral action

$R = diag(R_1,...,R_N) \in \Re^{Nl \times Nl}$. The stationary matrices L_u and L_w are calculated off-line from the input-output data via QR decomposition as described in [FDGV1998a].

A high level of formalization is the principle advantage of the data-based LQG method. The initial information includes an appropriate data-based model of the plant, the weighting matrices *Q* and *R* and the length of the optimization horizon *N*. The output of the method is the control law (4.7). The subspace identification is the ideological core of the method. It allows to estimate the sequence of the state vectors directly from the input-output data. Then the optimization problem (4.6), which does not require the prior knowledge of the matrices *(A,B,C,D)*, is solved. Hence, there is no need for an explicit Kalman's filter and, as a result, there are no problems with obtaining the filter and initializing it. While the conventional LQG method includes three steps: identifying an analytical model, synthesizing an LQ controller and a Kaman's filter, the data-based LQG "identifies" the control law (4.7) in one off-line step. The data-based LQG is obtained with the methods of linear algebra rather than by solving the two-point-boundary problem. In spite of this, in [FDVG1998a] the equivalence of the conventional LQG and the data-based LQG has been proved for (1) the white inputs and for (2) the infinite horizon $N \rightarrow \infty$.

The data-based LQG algorithm is capable of adapting to the plant evolution if, as soon as the control law becomes inadequate, a new open-loop data-based model is obtained and the control law is re-designed. A considerable drawback of such an approach is that it interrupts the normal plant operations. There can be two solutions to this problem.

First, if the plant evolution is implied by some known change in its structure or/and parameters, an updated data-based model can be obtained with the DVP, to which these known changes have to be introduced first. Then the control law is to be re-designed off-line, tested with the DVP, and applied to the actual plant.

Second, should the cause of the plant evolution be unknown, the closed-loop modification of the data-based LQG can be exploited [FDVG1998b]. It allows to obtain and to use the data measured in the closed-loop if the control law is known.

Since the data-based LQG implicitly involves the subspace identification, the requirements to the data-based model, which is used for the control system design, are the same as for the subspace identification method[18] in the sense of the persistence of excitation, number of samples, and spectrum characteristics of the input signal. An additional requirement to the mixed stochastic-deterministic subspace method is that the state and the output noise have to be uncorrelated and at least one of them must be non-zero [VD1996].

---

[18] The original article [FDVG1998a] does not explicitly refer to these requirements

A fundamental limitation of the method is that the stability of the control system cannot be guaranteed due to the finite horizon. Indeed, since the horizon is finite, it is impossible to say whether J in (4.6) remains finite as time tends to infinity (i.e. stable). Hardly can one rely only on the asymptotic properties of the data-based LQG.

As a consequence, the comprehensive testing of the designed control system in various regimes with varied parameters is necessary. The problem of testing is greatly simplified if the DVP of the plant is used because there is no need to have either the analytical model or the physical plant (as far as the DVP is considered as a good or "true" model of the actual plant).

The author of this dissertation has successfully applied the data-based LQG method in conjunction with the DVP method for controlling a double-link pendulum. The results have been published in [KT2003].

4.2.2.2. Iterative identification and control

The iterative identification and control (the batch stepwise adaptation) methods [Ite2002, TS1999] ideologically belong to the data-based group of methods despite their implementation being very close to the group, which requires explicit identification.

Unlike the conventional adaptive control methods, the identification of the plant and the control system re-design (tuning) are separated in time. The dynamics of the adaptation loop is assumed to be much slower than the dynamics of the control loop. This helps to avoid the notorious instability problems of the conventional adaptive structures. Also, the design complexity is reduced because the stages of the model identification and the control design are performed separately.

At each step of the adaptation process the plant model is updated and made more precise using the actual input/output data. The following two types of experiments are conducted [Ite2002]:

- The identification experiment to identify a fairly simple nominal analytical model of the plant, which describes the dominant processes;
- The validation experiment to obtain and to identify the uncertainty set, i.e. the difference between the plant and the analytical model.

The robust control system is designed in such a way that it would handle the nominal model along with any other model belonging to the uncertainty set. The transition steps from one controller to another are taken "cautiously", i.e. the updated controller is designed to stabilize both the previous and the updated models of the plant. In the context of this dissertation, the advantages of the close loop identification are also exploited (see Section 4.2.1.1).

A disadvantage of the iterative identification and control methods is that the convergence of the adaptation procedure cannot be guaranteed. Besides, when the DVP is used, an estimate of the model uncertainty set obtained with the DVP (generated data) will differ from an estimate that would be obtained using the real plant (real data). Hence, since the DVP-based uncertainty set is merely a difference between the DVP and the identified analytical model, it will not help to insure the robustness of the control system for the actual, real, plant. However, if the DVP can be assumed to be a very accurate representation of the plant, then the potential of the iterative identification and control methods may be very useful.

Nowadays the iterative identification and control methods are a very active research topic worldwide [Ite2002, TS1999]. It is hoped that the methods will help to adjust and to improve the existing industrial controllers without interrupting their operation. Employing the DVP is of advantage because it enables to accomplish the first step of the adaptation process and to test an initial control system on the DVP ***without*** any reference either to the analytical model or the real plant. This opens the new perspectives for introducing the iterative identification and control methods into the engineering practice.

4.3. Implementation of the method using the contemporary DVP and CACSD systems

This section is focused on implementing the proposed development method using the existing DVP and CACSD systems. As the proposed method is open, it does not depend on a specific vendor and can be implemented with a variety of computing tools available in the market. The non-standard elements of the method (special algorithms and functions for the DVP manipulation, identification and control system design) are to be implemented with the functionality and internal programming capabilities of the DVP and CACSD systems.

The most advanced and widely used DVP and CACSD systems are analyzed in the following two subsections from the point of view of the DVP-based control system development method.

4.3.1. DVP systems

Constructing and simulating the DVP of the plant can be accomplished in a number of DVP/CALS systems. The most widely used systems are: IGRIP/ENVISION of the D5 and V5 families [Das2005, Del2005], LMS/VirtualLab [LMS2005], MSC SimOffice (MSC.ADAMS, MSC.Nastran etc.) [MSC2005], Unigraphics NX (former I-deas) [UGS2005], ANSYS [ANS2005], and many more[19].

The DVP/CALS systems IGRIP and ENVISION[20], which have been extensively exploited in this dissertation, are developed by Delmia (USA) [Del2005], a branch of Dassault Systèmes (France) [Das2005]. They are intended for modeling mechanical and mechatronic systems, for example, robots and manipulators, as well as entire manufacturing lines. For example, the DVP/CALS systems make it possible to design and to simulate assembly processes, spot and arc welding, cutting, painting, ergonomics, VP calibration, to perform the collision analysis, off-line programming with translating the code from the internal programming language to the target controller language (for instance, the RAPID language found in the ABB robots). IGRIP and ENVISION are capable of solving the direct and inverse problems of kinematics and dynamics.

CATIA is the main internal format for representing the VP in the Dassault Systèmes products (V5) [Das2005]. The CATIA format has rather strong positions in the CAD community. The LMS/VirtualLab and MSC SimOffice systems also support CATIA [LMS2001, MSC2005]. Furthermore, the Dassault Systèmes products are capable of converting CATIA models into the STEP format and vice versa [Das2005].

STEP (ISO 10303: Standard for Exchange of Product Data) [ISO2005, Mas2003] is an international standard defining the means for representing and exchanging the information at all stages of the technical system lifecycle between heterogeneous CAD/CAE/CAM and CALS systems.

Currently the IGRIP/ENVISION modules of the D5 family are being gradually integrated into the latest product family V5. However, since the emphasis has been shifting towards the product lifecycle management (PLM), the V5 system does not have its own dynamics module. Currently Dassault Systèmes co-operates with the LMS and MSC Software products via CATIA to enable the advanced dynamic simulation in V5 using LMS and MSC dynamics engines [LMS2005, IBM2005].

---

[19] There are systems, which are more focused on the finite element method, for example, ABAQUS [www.hks.com], LS-DYNA [www.lsdyna-portal.com], COMSOL Multiphysics [COM2005]. The MathWork's Simulink/SimMechanics along with the relevant toolboxes (e.g. Virtual Reality toolbox) can be classified as a DVP system, though, with somewhat less extended capabilities of integrating with PDM systems (so far the "assemblies" can be imported only from SolidWorks) [MLAB2005]

[20] The major difference between IGRIP and ENVISION is that IGRIP has an extensive library of robots

Despite some differences in the application domains, all DVP systems, which are mentioned at the beginning of this sub-section, have very similar capabilities. All of them are intended for modeling rigid and elastic (depending on actual configuration of the DVP system) mechanical systems, i.e. their geometry, kinematics and dynamics. LMS/VirtualLab developed by LMS (Belgium) is able to model acoustics and vibrations [LMS2005]. MSC SimOffice is well-known for its modeling of technical systems with several physical phenomena, for instance, mechanical, electric, hydraulic and so on [MSC2005, Kro1999]. There exist other types of DVP systems. Thus, COMSOL Multiphysics/MEMS [COM2005] and FEMAS [FEM2002] handle the micromechanical systems, for which the gravity force is not dominant in comparison with the electrostatic and other micro-world-specific forces.

As a rule, the DVP/CALS systems support such standard formats as CATIA and STEP and have extensive means for integrating with the underlying PDM infrastructure (the VP or the digital mock-up according to the adopted terminology).

Let us formulate the main requirements to the functionality of the DVP system in the view of the proposed DVP-based control system development method. The efficiency and practical applicability of the proposed method strongly depends on the seamless information integration between the control development stage and other stages of the technical system lifecycle. The basic requirement is that the DVP system has to provide extensive means for such integration.

As far as the mathematical aspects of the DVP are concerned, apart form automating the construction of the equation of dynamics, a DVP system has to have (for mechatronic systems) the modules for solving the direct and inverse problems of kinematics and the module for solving the direct problem of dynamics (a selection of numerical integration methods capable of solving stiff equations and the equations with the computationally hard subintegral function). The inverse dynamics module may be useful for determining the stationary points of the modeled system. The numerical methods should allow to control such parameters of the numerical process as residuals, the absolute and relative errors, the condition number and the integration step size. It has to be possible to access the generalized coordinates of the modeled system, for example, the joint positions and velocities. The 3D dynamic visualization is the key element of the phenomenological description of the system by the DVP as it facilitates the man-machine interaction.

The simulation control program enables to take into account the external forces and torques. Their nature can be related to the reference command, control system actuator, forces that are difficult to account for in the standard DVP model (nonlinear air drag, brakes), and external disturbances (noise, vibration).

In IGRIP D5 the simulation control programs can be written in the dedicated high-level programming language[21], which contains such instructions as defining the simulation step size, setting the system to the initial state, simulating the system under the influence of the gravity and other external forces, performing file input/output and so on. There is a library of mathematical routines (`sin, cos, sqrt` etc.). The external forces and torques can be set and the generalized coordinates of the system can be read at each step of the simulation process. The integration methods include Runge-Kutta, predictor-corrector, and the implicit methods for stiff equations with controlling the absolute and relative errors as well as the integration step size. In sum, this functionality is sufficient for the accurate modeling of both the open loop and closed loop dynamic systems.

Our experience of using IGRIP D5 has indicated a good quality of the implemented numerical methods. This has been proved by modeling the same systems in IGRIP and MATLAB. However, it would be desirable to have a greater control over the numerical methods in IGRIP in terms of accessing the implementation details of the numerical methods, estimating the condition and the dynamics of residuals.

4.3.2. CACSD systems

The implementation of the final stage of the proposed development – designing the control system – is relatively uncomplicated. First, the input information for the CACSD systems is a formalized dynamic model. This helps to abstract from the nature and the application domain of the plant. Second, the CACSD systems are rather well developed and have a longer history of evolution than the DVP systems [Gru2001].

The outcome of the simulation experiment in the DVP system is the data-based dynamic model of the plant in the desired work regimes. This model can be used as a basis for the control system design with one of the two groups of methods described in the previous sections of this chapter. The simplest way to import the data-based model into the CACSD system is by a text file. If the DVP-based development method is to be commercialized, this phase can be improved by describing the data-based model in XML (Extensible Markup Language) [W3C2005] or STEP [ISO2005, MAS2003, Gru2001], [VHJ1996].

The essential functionality of the CACSD must include: file input/output, standard data processing (statistics, Fourier transform, filtering), 2D-3D visualization of the computing

[21] GSL (graphical simulation language) is used in IGRIP D5. To our knowledge, at present the V5 system lacks such a convenient, flexible and important tool as GSL

results, linear algebra, differential equations solvers, optimization, standard methods of the control system design (frequency and time domain analysis, LQ, Kalman filter, robust control), identification (in the frequency and time domains), and programming / scripting capabilities.

The most comprehensive and widely used implementation of the mathematical methods of control and identification is available in the CACSD MATLAB/Simulink (The MathWorks, USA)[22]. MATLAB/Simulink is a de-facto standard in the control engineering domain. The CACSD $MATRIX_X$ (National Instruments, USA) [MRX2005] is a recognized system as well. Its functionality is almost identical to MATLAB/Simulink's. The main disadvantage of the large commercial CACSD is their cost (thousands of euros) and vendor dependency. The CACSD Scilab / Scicos [SLAB2005] has the same capabilities as MATLAB/Simulink or $MATRIX_X$, but it is distributed freely along with its source code[23].

Among other mathematical software suitable for general mathematical computations we note Maple [Map2005], Mathcad [MCAD], Mathematica [MATH2005] and freely distributed open-source Octave [OCT2005].

4.4. Summary

1. The DVP control system development method has been adapted to the selected class plants. The class includes the plants which are linearizable, quasi-stationary, and stable or stabilizable without using the analytical model and have lumped parameters;

2. The possibility of developing the desired control system for a plant, which belongs to the selected class, using the DVP-based method has been proved. The essence of the proof is demonstrating that there exist a coherent chain of modeling, identification and control system design methods, which lead to the desired solution of the control problem subject to the availability of the necessary information on the plant. It has been shown that the DVP-based control system development method can be implemented using the contemporary DVP and CACSD systems;

3. The specifics, limitations and rules of applying the existing mathematical methods of identification and control system design in the context of DVP have been systematically analyzed. The following two major groups of methods have been considered: the methods, which involve the intermediate identification of the analytical model, and the data-based

[22] Hardly can such an overwhelmingly complex system for technical computing as MATLAB/Simulink be called a CACSD. Here we mean a subset (the relevant toolboxes) of MATLAB/Simulink, which is used as CACSD

[23] Scilab has a translator of the MATLAB m-functions

methods, which do not require such identification. The ways of improving the quality and adequacy of the results have been discussed (e.g. the information capacity of the data-based model and the closed-loop identification);

4. The insufficiency of the information on the model uncertainty set and the lack of the formal MIMO control system design methods, which would not require the explicit knowledge of the uncertainty set structure, have been indicated as a bottleneck of the proposed DVP-based method. Two robustness-related problems have been considered and a procedure for constructing the general form of the uncertain plant on the basis of multiple simulations in the DVP system has been suggested.

CHAPTER 5. APPLICATION OF THE DVP CONTROL SYSTEM DEVELOPMENT METHOD TO GANTRY CRANES

In this chapter a control system for fine positioning of a *gantry crane* (GC) load will be considered to illustrate the DVP-based development method and to prove its engineering applicability.

The GC is a representative example. On the one hand, it is a "classical" and sufficiently simple plant. It has a clear structure and intuitive physical behavior that do not overshadow the proposed development method. On the other hand, the GC is interesting due to its dynamics. The GC is a MIMO system with the DOFs varying in their stability type and controllability. Some nonlinear dynamic effects take prominence even when the GC phase coordinates remain within the work range.

There is a body of literature dedicated to the modeling and simulation of the GCs, which includes a handbook [Uns1976] on constructing cranes and analyzing their dynamics and a fundamental English language literature review [ANM2003] containing 167 references on the modeling and control of cranes. Nevertheless, the GC control problem is known to have not been completely solved to date [ANM2003, Rob2003].

The model of the GC referred to in this chapter is adopted from article [SS1982], which has become standard in the domain of the GC modeling and control. The model structure and parameters correspond to the real GC.

Let us introduce the terminology, which is used further in the chapter. There are four types of the GC models:

1. The ***reference model*** of the GC, its structure and parameters, are the basis for the control system development and the DVP of the GC (see Section 5.1);
2. The ***analytical model*** of the GC is the mathematical description of the reference model dynamics in the form of the nonlinear and linearized ordinary differential equations. This model can be obtained analytically from the first principles or by means of identifying the data-based model (see Section 5.5);
3. The ***data-based model*** of the GC is the mathematical description of the reference model dynamics in the table input-output form (see Section 5.6.2). By the proposed development method, the data-based model is generated by simulating the GC DVP rather than obtained with a GC physical embodiment;
4. The ***dynamic virtual prototype*** (DVP) of the GC. It is built according to the reference model (see Section 5.6.1).

This chapter has three major parts:

- Description of the GC reference model. The importance of the GC control. The control problem formulation. Review of the existing methods of the GC control. Efficiency analysis of the proposed DVP development method;
- Derivation of the analytical nonlinear and linearized dynamics models of the GC to be compared with the GC DVP simulation. Qualitative analysis of the analytical models and the literature overview. Simulation of the analytical models in the GC reference work regimes. Assessment of how well the linearized model approximates the nonlinear GC dynamics;
- Development of the GC control system using DVP. The order of the material follows the main stages of the proposed development method:
    1. Constructing the GC DVP in accordance with the GC reference model;
    2. Simulating the GC DVP to generate the GC data-based model, which represents the GC dynamics in the reference work regime. Identifying the GC (linear) model;
    3. Designing the control system based on the identified model.

The attention is paid to the following three aspects:

1. Modeling and control design of the GC;
2. Comparing the results of the existing and the proposed DVP development methods;
3. Analyzing the specifics of the DVP development method, assessing its efficiency, and determining the area of applicability.

Hardly is it possible to understand completely the control problem without thorough plant research. Hence, the material on the GC analytical modeling, the review of the existing approaches to the GC control, and the analysis of the factors affecting the dynamic model adequacy are presented at length to interpret correctly the results, to show the diversity, depth, and nontriviality of the problems addressed to in this dissertation.

5.1. The gantry crane reference model

Cranes are used for the cyclic transportation of loads inside a bounded work space [Uns1976]. A crane consists of a hoisting mechanism (ropes and a hook or a spreader) and a support mechanism (trolley-girder, trolley-jib, or a boom). The rope-hook-load assembly is suspended from the support mechanism. Cranes can be classified depending on the type and the number of the suspension point DOFs. There are gantry (overhead), tower (rotary), and boom cranes [ANM2003, Uns1976]. The gantry cranes (GC), which are utilized in the containership industry, are considered as an example in this dissertation.

The GC reference model has been adopted from [SS1982]. The structure of the model is shown in Figure 5.1; the parameters are given in Table 5.1. In general, the parameters can be either obtained from the GC specification or found experimentally [Uns1976]. The model in [SS1982] does not take the viscous friction into account. So, the viscous friction coefficients have been selected as described in Appendix 1. The detailed analysis of the model will be presented in Section 5.5.2.

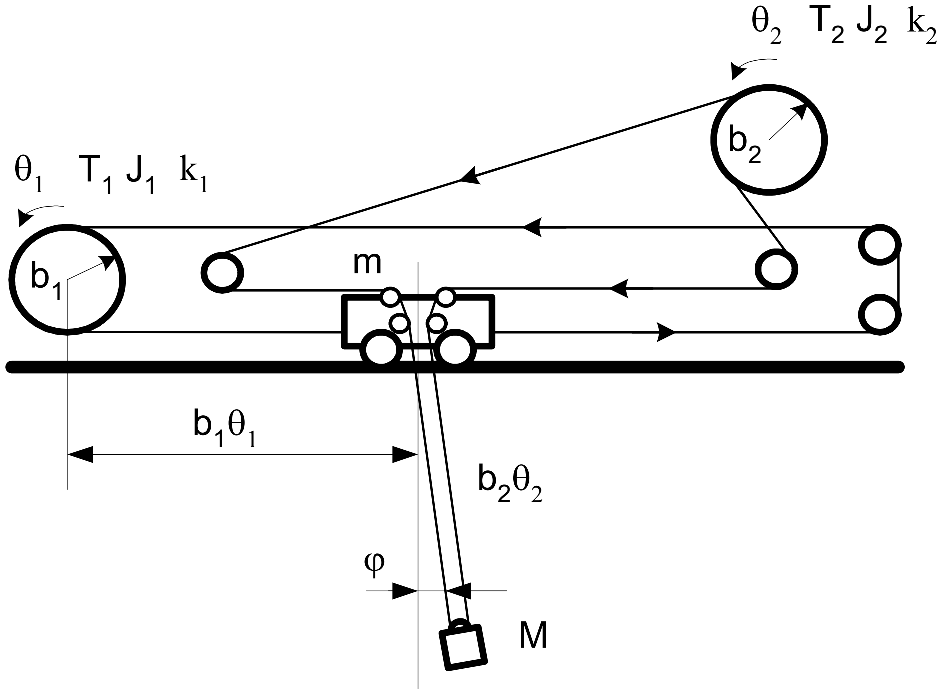

Figure 5.1. Gantry crane reference model (notation is given in Table 5.1)

The GC reference model is an electromechanical system with the following three mechanical DOFs: the rotation angle of the trolley drive motor θ_1, the rotation angle of the hoist drive motor θ_2, and the load angular displacement φ. The GC has two inputs T_1, T_2 and three outputs θ_1, θ_2, φ.

There are a number of parameters associated with these DOFs (the units are given in square brackets):

- The rotation angle of the trolley drive motor: θ_1 [*rad*]. The linear horizontal position of the trolley: $b_1\theta_1$ [*m*], where b_1 [*m*] is the equivalent radius of the trolley drive motor drum, which is reduced to the motor side. The total moment of inertia of the trolley drive motor, the brake, the drum, and the reduction gears: J_1 [*kg·m²*]. The viscous friction of the trolley drive: k_1 [$kg \cdot m^2 \cdot s^{-1}$]. The trolley mass: m [*kg*]. The driving torque generated by the trolley drive motor: T_1 [*N·m*].

Table 5.1. Parameters of the gantry crane reference model[1]

	Parameter	Notation	Value
Trolley drive	Trolley mass [*kg*]	m	$6.0 \cdot 10^3$
	Total moment of inertia [$kg \cdot m^2$]	J_1	3.75
	Equivalent drum radius, [*m*]	b_1	$1.44 \cdot 10^{-2}$
	Maximum motor torque, [$N \cdot m$]	$T_{1\,max}$	$1.03 \cdot 10^3$
	Maximum linear trolley velocity, [*m/s*]	$v_{1\,max}$	2.5
	Viscous friction of the trolley drive [$kg \cdot m^2 \cdot s^{-1}$]	k_1	0.92
Hoist drive	Total load mass [*kg*]	M	$4.25 \cdot 10^4$
	Total moment of inertia [$kg \cdot m^2$]	J_2	78.5
	Equivalent drum radius [*m*]	b_2	$1.22 \cdot 10^{-2}$
	Maximum motor torque [$N \cdot m$]	$T_{2\,max}$	$1.09 \cdot 10^4$
	Maximum linear trolley velocity [*m/s*]	$v_{2\,max}$	1.0
	Viscous friction of the hoist drive [$kg \cdot m^2 \cdot s^{-1}$]	k_2	0.92
	Viscous friction of the load angular displacement [$kg \cdot m^2 \cdot s^{-1}$]	k_3	$3.9 \cdot 10^5$
	Maximum rope length [*m*]	L_0	28

- The rotation angle of the hoist drive motor: θ_2 [*rad*]. The rope length: $b_2\theta_2$ [*m*], where b_2 [*m*] is the equivalent radius of the hoist drive motor drum, which is reduced to the motor side. The total moment of inertia of the hoist drive motor, the brake, the drum, and the reduction gears: J_2 [$kg \cdot m^2$]. The viscous friction of the hoist drive: k_2 [$kg \cdot m^2 \cdot s^{-1}$]. The mass of the load, the ropes and the spreader: M [*kg*]. The driving torque generated by the hoist drive motor: T_2 [$N \cdot m$].
- The load angular displacement: φ [*rad*]. The viscous friction of the load angular displacement: k_3 [$kg \cdot m^2 \cdot s^{-1}$].

The GC reference model includes the viscous friction (unlike the model in [SS1982]) and the moments of inertial of the drives (unlike the model in [AT1987]).

An important source of information regarding the construction and the control system of GCs is the national and the international standards and regulations. A theoretically flawless

[1] In practice, there are more parameters and constraints, e.g. thermal constraints, safety regulations and so on

control algorithm or a dynamic model of a crane may become impractical if the standard requirements are not met. Here we refer to, for example, the national Finnish standards on safety, design, and requirements for electrotechnical equipment [SFS2004], the rules for the design of hoisting appliances [SFS1977a, SFS1977b], along with a set of international standards such as [ISO1981, ISO1995a, ISO1995b].

There are several factors limiting the reference model level of details. The ***first*** factor is the incomplete information about the structure and parameters of the model. Thus, we have no information about the elasticity and the Coulomb friction of the GC. The ***second*** factor is the type of the DVP system, which is IGRIP in our example. This DVP system allows to model the mechanical phenomena directly. Other phenomena (electromechanical, aerodynamical) have to be described in IGRIP analytically. Although, for example, ADAMS [MSC2005] and LMS/VirtualLab [LMS2005] are capable of direct modeling some non-mechanical phenomena (no explicit analytical expressions are needed). The ***third*** factor is that the model does not have to be more detailed than it would be necessary for the adequate description of the phenomena, which are relevant to the control loop at the level of macro variables [PS2003, DK2001, Hyo2003].

5.2. The control problem formulation

The dynamic behavior of the GC can be very complex due to the specifics of its construction structure [ANM2003, EKM2002, MN2003, OHH1999]. An external excitation of the suspension point may lead to spatial load oscillations in the plane the trolley motion, and out that plane. The oscillations can be periodic, aperiodic or chaotic. Due to inertia, moving the trolley and the crane inevitably generates the load oscillations, which decay very slowly because of the extremely low construction rigidity and damping.

The GCs are a bottleneck in the containership transportation chain [Rob2003, ANM2003]. The productivity of the modern GC is about 30 cycles per hour. In future, the cycle rate is planned to be raised by about one third. A large container ship carries several thousand containers. If two or three GCs are working in parallel in three shifts, it will take about 24 hours (practically, several days) to unload the ship. Therefore, solving the task of minimizing the GC work cycle to improve the productivity of the transportation chain has a very solid economic motivation.

The duration of the work cycle depends on a number of factors such as [Uns1976] the GC dimensions, maximum accelerations and velocities, trajectory types, combination of motions, work process organization, operator qualification etc. From the point of view of dynamics, reducing the cycle duration by increasing the work velocities and accelerations contradicts keeping the desired precision of the load positioning. Specifically, the more intensive the GC operations are, the larger the overshoot and the amplitude of the residual oscillations become resulting in longer cycles and less safety.

The above-mentioned contradiction can be resolved by means the construction modifications of the GC and the dynamic control. The later balances the speed and the precision of the GC operations. There are two approaches to controlling the GC.

First, the load transportation, which is an inverse dynamics problem, is formulated in an absolute coordinate system with a fixed origin. The problem is solved using the optimal control methods. Thus, in [SS1982] the entire trajectory of the work cycle is divided into several pieces: hoisting, trolley translating, lowering and so on. The optimal control and trajectory shape are designed to minimize a quality functional, which is related to the duration of the cycle, to make the junctions of the trajectory pieces smooth, and to keep the zero residual oscillations at the target position. The nonlinear equations of the GC dynamics and Pontryagin's principle are used to find the optimal control.

Second, by analogy with robotics [Yur2001], the rough quick and the fine slow motion phases are introduced. On the one hand, the rough motion is accomplished by the GC operator along predefined reference trajectories. The reference trajectory may be chosen to minimize the excitation of the oscillations, but large oscillations are not suppressed. In fact, the large swing at the start of the motion is unavoidable because the only way to set the load in motion is to create a nonzero deflection angle. The same rule applies at the end of the motion. One forward swing is allowed, though not necessarily jeopardizing the positioning precision, to stop the load. On the other hand, the residual oscillations, which become prominent at the end of the rough motion phases and during the fine positioning, are avoidable[2]. The problem of suppressing the residual oscillations is formulated in the coordinate system, which is relative to the reference (desirable) trajectory, and is solved using the standard automatic control methods.

The second approach leads to suboptimal solutions. But the first approach, in spite of being mathematically more formal, is not flawless either. Firstly, its solution may be inflexible and sensitive to external disturbances because the motion goes along a pre-computed trajectory (feedforward control if the final state is fixed [LS1995]). Secondly, the solution might not be robust to model parameter uncertainty [ANM2003]. Thirdly, the

[2] The fact that only the small and avoidable oscillations are suppressed corresponds to the thixotropy principle

solutions obtained by the first approach are almost never put in practice [ANM2003, Ver2002]. This happens because the first approach, at least implicitly, aims at a fully automated crane with no human presence. It contradicts the strict safety regulations, with which are very difficult to comply, if there is no human operator. Moreover, the GC operators experience psychological discomfort when the crane is fully automated and its behavior does not completely obey their commands [Rob2003]. It is known, though, that the experienced GC operator does not perform worse than any up-to-date GC control system [*ibid*]. So, it is difficult to justify the additional costs related to the control system.

As a result, many GCs, which do have the control systems, are still controlled manually and the control systems are used exclusively for the fine load positioning. In [ANM2003, Ver2002, Rob2003] it is clearly pointed out that one needs such a GC control system that would not substitute the human operator, but would be of assistance in some of the work regimes, for instance, during the fine positioning.

A detailed research by Konecranes Oy [Rob2003] has shown that the fine positioning takes, on average, from 43% to 53% (up to 80% for the inexperienced operator) of the total work cycle period. According to [Uns1976], it has been estimated that the fine positioning control system may reduce the work cycle by as much as 20%. Hence, suppressing the residual load oscillations is a practical and efficient way of reducing the work cycle period.

To sum up, the fine load positioning takes a large proportion of the work cycle period and, therefore, strongly affects the productivity of the GC. There is a practical need in the fine positioning control system, which suppresses the residual load oscillations, to assist the human operator, to decrease the work cycle period, to reduce the vibrations of the GC, and to improve the safety of operations. Mathematically the control problem has to be formulated in the coordinate system that is relative to the load target position. The control system must be robust to the GC parametric uncertainties, which mainly include a variety of the container mass and the changing rope length.

Let us formulate the GC fine positioning control problem.

Move the load along the desired trajectory in minimum time and with minimum residual oscillations along the trajectory and in the proximity of the target position subject to the limited velocities, accelerations and driving torques defined by the construction of the crane.

5.3. Analysis of the GC control methods

From the point of view of the automatic control, the GC is difficult to control because [ANM2003, Rob2003, SS1982]

- The GC is a very lightly damped system;
- Due to the GC structure, most of the known crane control systems suppress the load oscillations by manipulating the trolley position or/and the rope length. In those cases the control is exerted rather ineffectively with respect to the load. The longer the rope is, the more difficult it is to affect the load. Consequently, the control quality decreases while the size of the crane increases;
- The GC is a nonlinear system. Its parameters vary both within a work cycle (the rope length, motion with and without a container) and between cycles (the container mass).

Consider the main methods of reducing or suppressing the load oscillations [Uns1976, ANM2003, Pat2002]. The methods can be classified depending on the place where control is exerted [Uns1976]:

- Influencing the excitation of the load oscillations;
- Influencing the sensitivity of the GC to the load oscillations.

Depending on the implementation, the methods can be classified as follows: [ANM2003]:

- The methods that require changing the construction and the structure of the GC;
- The methods of the dynamic control.

The methods that require changing the GC construction to reduce the excitation of the load oscillations imply limiting the maximum velocities, accelerations and drive torques, selecting special motors, brakes etc.

The methods that require changing the GC construction to reduce the sensitivity of the GC to the load oscillations include employing oscillation absorbers, increasing the elasticity of the construction (the V-type rope reeving), damping oscillations with an additional side rope (one end of the rope is connected to the spreader while the other is controlled by the active oscillation absorber), employing the motors with the special speed-torque characteristic to increase damping.

The dynamic control can be accomplished by either the human operator or the automatic control system. The rules for the GC operators are quite natural. For example, jerky operations are forbidden and motion of the load has to occur with the shortest rope length [Ver2002]. A set of universal principles can be used for the GC dynamic control [Yur2001]:

1. The principle of coordinating the free and the forced motions. For example, the acceleration and deceleration phases can be synchronized with the half-period of the load oscillations [ANM2003];

2. The principle of separating the rough quick and the fine slow motions;

3. The principle of coordinating the motion of several DOFs (in time, in space, and mechanically). Thus, if the load is hoisted when the angular displacement is maximal, or the load is lowered when the angular displacement is minimal, then the oscillations will be reduced. The same effect can be achieved if the trolley is accelerated when the load is ahead of the trolley, and the trolley is decelerated when the load is behind the trolley. This principle is widely used by the GC operators [Uns1976];

4. The principle of combining several types of control. For instance, the GC can be controlled by the human operator, which is assisted by the automatic control system for suppressing the residual load oscillations;

5. The principle of introducing several hierarchical levels of control. For example, the human operator defines the reference trajectory (at the high level) and the automatic control system drives the GC along this trajectory (at the low level) keeping the residual oscillations small [ANM2003, Uns1976];

6. The principle of coordinating the control system and the associated hardware. For instance, the requirements for the spreader positioning precision can be relaxed if the spreader is equipped with the guide pins to assist positioning the spreader [Uns1976];

One additional principle is

7. The principle of thixotropy [Fre1935], i.e. relaxing and passing through the external forces, which the control system is unable to tackle (e.g. to avoid the system destruction).

These principles have been employed for defining the control problem in Section 5.2.

We have to mention the active control system of the spreader position that has been proposed by Konecranes Oy [Pat2002, KCI2001, Rob2003]. The system makes use of the auxiliary crossed ropes, which enable to apply the oscillations compensating force to the spreader directly. This method allows to reduce the excitation of the load oscillations, to decrease the hoist drive load, and to improve the load positioning precision. Despite the fact that the efficiency of the control system degrades as the rope length increases, the achievable positioning precision of the GC is remarkable. The GC described in [KCI2001] has the spreader positioning precision of ±150 *mm* and the angular spreader tilt of ±3° with the maximum rope length of 55 *m*. The feature of the system is that it is capable of suppressing the load oscillations both in and out of the trolley motion plane. There is no doubt that the developing of the automatic control for such a system is not a trivial task.

A large number of the control systems for suppressing the GC load oscillations have been proposed [ANM2003]. Furthermore, there are control methods for the applications similar to the GC, for example, the control systems for cargo helicopters [PLL2001] and vibration machines [ABB2001].

The GC control methods can be classified into the open-loop and the closed-loop (feedback) methods. Virtually every existing approach of the automatic control has been exploited for controlling the GCs. There are input-shaping, PID, robust (including μ-controllers [KZ2001]), LQ and Pontryagin-principle-based optimal, adaptive, nonlinear, fuzzy and neural controllers. However, despite the intensive research, the problem of the GC load control has not been completely solved for the time being. The first reason is the complexity of the problem. The second one is that not always does the academic research exactly correspond to the practical needs.

5.4. Efficiency of the DVP control system development

The feature of the GC development and manufacture is that a new crane is (1) produced to order with the precisely (2) customized parameters (size, carrying capacity, wind resistance) in (3) a short time (12-14 months) [Rob2003]. To fulfill the requirements of the customer, the base model of the GC is adjusted. The adjustments affect the dynamic model and the control system of the crane. If the control system is being developed on the basis of a separate analytical model, it can be difficult to update the model and to re-design the control system in the short period of time. Hence, assuming that the customization adjustments are first introduced to the virtual prototype of the GC, it makes sense to utilize the proposed DVP development method to update the GC control system ***automatically***. The DVP control system development method helps to ***reduce the development time*** and to ***eliminate the information barrier*** between the virtual prototype and the corresponding control system.

The more detailed the dynamics model is, the larger it becomes. If the analytical model of GC is extended to include such phenomena as Coulomb friction, brakes, auxiliary ropes, dampers, non-idealized motors, elasticity of the drives, non-planar load oscillations, and the double-pendulum-like rope-spreader-load assembly (see Section 5.5.2), then the model will turn out to be overloaded with details and practically worthless.

For example, the equations of dynamics of the GC reference model contain about ***ten*** terms (Section 5.5.1). Some five extra DOF can be added to the model to obtain a more realistic dynamics (3D motion of the double-link pendulum, electrical DOF for both drives). The model with eight DOFs will have ***hundreds*** of terms, i.e. roughly $(8/3)^3 \approx 20$ times more than the model with three DOFs (Section 2.4). Hardly is it possible to construct such a model without the computer.

In contrast, if the necessary information is available, the GC DVP could be ***more detailed*** than the analytical model without loss of practical usefulness because the ***routine work is minimized*** and there is ***no need to tackle a large number of the bulky equations***. The detailed DVP may provide a ***better insight into the system nonlinear dynamics***, of course, at the cost of intensive computations.

Compared with the conventional analytical modeling (knowledge-intensive) and the real plant experiments (data-intensive), the DVP-based method (simulation-intensive) is ***less knowledge-intensive*** and it ***does not require expensive experiments***.

Moreover, if the DVP development method is applied together with the conventional analytical modeling or the real plant experiments, the DVP method provides a relatively ***independent way of validating the results***. The author has enjoyed this feature while working out the numerical examples described in the following sections and in [KT2003].

To summarize, the DVP systems appear to be appropriate for developing the GC control systems.

5.5. The analytical model of the gantry crane

5.5.1. Derivation

The dynamics of the GC reference model can be described by Lagrange's equations of the second type because the constraints are holonomic[3] [Tsa1999]. Basically, the GC reference model is equivalent to a serial robotic manipulator with three mechanical DOFs.

Let the vector of the generalized coordinated be $q = (\theta_1, \theta_2, \varphi)$. Then, the kinetic energy of the system is expressed as:

$$T = \frac{1}{2}J_1\dot{\theta}_1^2 + \frac{1}{2}J_2\dot{\theta}_2^2 + \frac{1}{2}m\left(b_1\dot{\theta}_1\right)^2 +$$
$$+\frac{1}{2}M\left\{\left(b_1\dot{\theta}_1 + b_2\dot{\theta}_2\sin\varphi + b_2\theta_2\dot{\varphi}\cos\varphi\right)^2 + \left(b_2\dot{\theta}_2\cos\varphi - b_2\theta_2\dot{\varphi}\sin\varphi\right)^2\right\}.$$

[3] Holonomic constraints mean that there are no velocity constraints. If there were nonholonomic constraints, Lagrange's equations of the first type would have to be used

The first term corresponds to the kinetic energy of the trolley drive; the second term is due to the kinetic energy of the hoist drive; the third term describes the kinetic energy of the trolley; finally, the forth term is the kinetic energy of the load. The expression in the curled brackets is the sum of squares of the load linear velocity projections on the vertical and the horizontal axes in the fixed coordinate system. The velocity is composed of the trolley linear velocity $b_1\dot{\theta}_1$ and the load linear velocity along the rope (hoisting/lowering) $b_2\dot{\theta}_2$ and perpendicular to the rope (load oscillations relative to the trolley) $b_2\theta_2\dot{\varphi}$. The potential energy of the system (due to the load) can be written as: $P = -Mgb_2\theta_2\cos\varphi$.

Lagrange's equations of the second type have the following form:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i, \quad i = \overline{1,3}, \tag{5.1}$$

where $L = T - P$ is the Lagrangian, q_i is the i-th generalized coordinate, Q_i is the generalized non-potential force for the i-th generalized coordinate (in our case it has the units of torque [$N{\cdot}m$]). For the non-potential forces we have:

$$\begin{aligned} Q_1 &= T_1 - k_1\dot{\theta}_1, \\ Q_2 &= T_2 - k_2\dot{\theta}_2, \\ Q_3 &= -k_3\dot{\varphi}, \end{aligned} \tag{5.2}$$

where $k_i\dot{q}_i$ the torque of the viscous friction for the corresponding generalized coordinate. Differentiating the Lagrangian with respect to the generalized velocities, we have:

$$\begin{aligned} \frac{\partial L}{\partial \dot{\theta}_1} &= J_1\dot{\theta}_1 + mb_1^2\dot{\theta}_1 + Mb_1\left(b_1\dot{\theta}_1 + b_2\dot{\theta}_2\sin\varphi + b_2\theta_2\dot{\varphi}\cos\varphi\right), \\ \frac{\partial L}{\partial \dot{\theta}_2} &= J_2\dot{\theta}_2 + Mb_2\left(b_2\dot{\theta}_2 + b_1\dot{\theta}_1\sin\varphi\right), \\ \frac{\partial L}{\partial \dot{\varphi}} &= Mb_2^2\theta_2^2\dot{\varphi} + Mb_1b_2\dot{\theta}_1\theta_2\cos\varphi, \end{aligned} \tag{5.3}$$

and then differentiating the result with respect to time, we obtain:

$$\begin{aligned} \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_1} &= \left(J_1 + (M+m)b_1^2\right)\ddot{\theta}_1 + \\ &\quad + Mb_1b_2\left(\ddot{\theta}_2\sin\varphi + 2\dot{\theta}_2\dot{\varphi}\cos\varphi + \theta_2\ddot{\varphi}\cos\varphi - \theta_2\dot{\varphi}^2\sin\varphi\right), \\ \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_2} &= \left(J_2 + Mb_2^2\right)\ddot{\theta}_2 + Mb_2\left(b_1\ddot{\theta}_1\sin\varphi + b_1\dot{\theta}_1\dot{\varphi}\cos\varphi\right), \\ \frac{d}{dt}\frac{\partial L}{\partial \dot{\varphi}} &= Mb_2^2\left(2\theta_2\dot{\theta}_2\dot{\varphi} + \theta_2^2\ddot{\varphi}\right) + \\ &\quad + Mb_1b_2\left(\ddot{\theta}_1\theta_2\cos\varphi + \dot{\theta}_1\dot{\theta}_2\cos\varphi - \dot{\theta}_1\theta_2\dot{\varphi}\sin\varphi\right). \end{aligned} \tag{5.4}$$

Taking the derivative of the Lagrangian with respect to the generalized coordinates yields:

$$
\begin{aligned}
&\frac{\partial L}{\partial \theta_1} = 0, \\
&\frac{\partial L}{\partial \theta_2} = Mb_2\dot{\varphi}\left(b_1\dot{\theta}_1 \cos\varphi + b_2\theta_2\dot{\varphi}\right) + Mgb_2 \cos\varphi, \\
&\frac{\partial L}{\partial \varphi} = Mb_1\dot{\theta}_1\left(b_2\theta_2\dot{\varphi}\sin\varphi - b_2\dot{\theta}_2 \cos\varphi\right) - Mgb_2\theta_2 \sin\varphi.
\end{aligned}
\tag{5.5}
$$

Substituting (5.2) through (5.5) into (5.1), we obtain:

$$
\begin{cases}
\left(J_1 + (M+m)b_1^2\right)\ddot{\theta}_1 + Mb_1b_2\left(\ddot{\theta}_2 \sin\varphi + \theta_2\ddot{\varphi}\cos\varphi + \right. \\
\qquad \left. + 2\dot{\theta}_2\dot{\varphi}\cos\varphi - \theta_2\dot{\varphi}^2 \sin\varphi\right) + k_1\dot{\theta}_1 = T_1, \\
Mb_1b_2\ddot{\theta}_1 \sin\varphi + \left(J_2 + Mb_2^2\right)\ddot{\theta}_2 - Mb_2^2\theta_2\dot{\varphi}^2 - \\
\qquad - Mgb_2 \cos\varphi + k_2\dot{\theta}_2 = T_2, \\
Mb_1b_2\theta_2\ddot{\theta}_1 \cos\varphi + Mb_2^2\theta_2^2\ddot{\varphi} + 2Mb_2^2\theta_2\dot{\theta}_2\dot{\varphi} + \\
\qquad + Mgb_2\theta_2 \sin\varphi + k_3\dot{\varphi} = 0.
\end{cases}
\tag{5.6}
$$

Hence, we have arrived to the equations given in [SS1982] with the difference that (5.6) accounts for the viscous friction.

Let us linearize [Zak2003] the nonlinear system of equations (5.6) about the operation (equilibrium) point of the load fine positioning system. The phase vector of the nonlinear system (5.6) is given by:

$$
X = \begin{pmatrix}\theta_1 & \theta_2 & \varphi & \dot{\theta}_1 & \dot{\theta}_2 & \dot{\varphi}\end{pmatrix}^T. \tag{5.7}
$$

The phase vector at the operation point can be written as:

$$
X^{(0)} = \begin{pmatrix}\theta_1^{(0)} & \theta_2^{(0)} & \varphi^{(0)} = 0 & \dot{\theta}_1^{(0)} = 0 & \dot{\theta}_2^{(0)} = 0 & \dot{\varphi}^{(0)} = 0\end{pmatrix}^T. \tag{5.8}
$$

Therefore, the phase vector of the linearized system is given by:

$$
x = X - X^{(0)} = \begin{pmatrix}x_1 & x_2 & x_3 & x_4 & x_5 & x_6\end{pmatrix}^T. \tag{5.9}
$$

The input and the output vectors of the linearized system are:

$$
\begin{aligned}
&u = \begin{pmatrix}T_1 & T_2 - Mgb_2\end{pmatrix}^T = \begin{pmatrix}u_1 & u_2\end{pmatrix}^T, \\
&y = \begin{pmatrix}x_1 & x_2 & x_3\end{pmatrix}^T.
\end{aligned}
\tag{5.10}
$$

Expression $u_2 = T_2 - Mgb_2$ means that at the operation point $X^{(0)}$ the hoist drive compensates for the gravity force applied to the load. Taking Taylor's expansion of (5.6) about $X^{(0)}$, denoting $b_2\theta_2^{(0)}$ by L_0, and truncating the terms of the second and higher orders, we obtain:

$$
\begin{cases}
\left(J_1 + (M+m)b_1^2\right)\dot{x}_4 + Mb_1L_0\dot{x}_6 + k_1x_4 = u_1, \\
\left(J_2 + Mb_2^2\right)\dot{x}_5 + k_2x_5 = u_2, \\
ML_0b_1\dot{x}_4 + ML_0^2\dot{x}_6 + ML_0gx_3 + k_3x_6 = 0.
\end{cases}
\tag{5.11}
$$

In order to describe the system in the standard state-space form [Zak2003], the first derivatives of the state variables in (5.9) have to be written explicitly. From (5.7) and (5.9), it follows that: $\dot{x}_1 = x_4; \dot{x}_2 = x_5; \dot{x}_3 = x_6$. The expressions for state variables $\dot{x}_i, i = \overline{4,5}$ can be obtained by solving the linear system of equations, which is derived from (5.11):

$$\underbrace{\begin{pmatrix} J_1 + (M+m)b_1^2 & 0 & ML_0 b_1 \\ 0 & \left(J_2 + Mb_2^2\right) & 0 \\ ML_0 b_1 & 0 & ML_0^2 \end{pmatrix}}_{S} \begin{pmatrix} \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{pmatrix} = \begin{pmatrix} u_1 - k_1 x_4 \\ u_2 - k_2 x_5 \\ -ML_0 g x_3 - k_3 x_6 \end{pmatrix}. \tag{5.12}$$

The solution[4] of system (5.12) is:

$$\begin{aligned} \dot{x}_4 &= S_{11}^*(u_1 - k_1 x_4) + S_{12}^*(u_2 - k_2 x_5) + S_{13}^*(-ML_0 g x_3 - k_3 x_6) = \\ &= x_3\left(-S_{13}^* ML_0 g\right) + x_4\left(-S_{11}^* k_1\right) + x_5\left(-S_{12}^* k_2\right) + x_6\left(-S_{13}^* k_3\right) + \\ &+ u_1 S_{11}^* + u_2 S_{12}^*, \\ \dot{x}_5 &= S_{21}^*(u_1 - k_1 x_4) + S_{22}^*(u_2 - k_2 x_5) + S_{23}^*(-ML_0 g x_3 - k_3 x_6) = \\ &= x_3\left(-S_{23}^* ML_0 g\right) + x_4\left(-S_{21}^* k_1\right) + x_5\left(-S_{22}^* k_2\right) + x_6\left(-S_{23}^* k_3\right) +, \\ &+ u_1 S_{21}^* + u_2 S_{22}^*, \\ \dot{x}_6 &= S_{31}^*(u_1 - k_1 x_4) + S_{32}^*(u_2 - k_2 x_5) + S_{33}^*(-ML_0 g x_3 - k_3 x_6) = \\ &= x_3\left(-S_{33}^* ML_0 g\right) + x_4\left(-S_{31}^* k_1\right) + x_5\left(-S_{32}^* k_2\right) + x_6\left(-S_{33}^* k_3\right) + \\ &+ u_1 S_{31}^* + u_2 S_{32}^*, \end{aligned} \tag{5.13}$$

where $S_{ij}^*, i, j = \overline{1,3}$ are the elements of the matrix S^*, which is the inverse of the matrix S in (5.12).

There are two methods of inverting the matrix *S*. The first method is to substitute the numerical values of the parameters into *S* and to use the numerical methods [KMN1989, BZK2000, GV1989, PTV1999]. The second method is to obtain the matrix S^* analytically. We can use the software for the symbolic mathematical computations, for example, MATLAB / Symbolic Toolbox [MLAB2005]. Exploiting function `inv` of MATLAB, we obtain:

$$S^* = \begin{pmatrix} \dfrac{1}{J_1 + b_1^2 m} & 0 & \dfrac{-b_1}{L_0\left(J_1 + b_1^2 m\right)} \\ 0 & \dfrac{1}{J_2 + Mb_2^2} & 0 \\ \dfrac{-b_1}{L_0\left(J_1 + b_1^2 m\right)} & 0 & \dfrac{J_1 + b_1^2(m+M)}{ML_0^2\left(J_1 + b_1^2 m\right)} \end{pmatrix}. \tag{5.14}$$

4 It is always possible to express the first derivatives of the state variables explicitly due to the holonomic constrains

Finally, the linearized stationary model of the GC in the state-space form is given by:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t), \\ x(0) &= x_0, \end{aligned} \tag{5.15}$$

where $A \in R^{n \times n}, B \in R^{n \times m}, C \in R^{l \times n}, x \in R^n, u \in R^m, y \in R^l$. In our case, *n=6* (the dimensionality of the state space), *m=2* (the dimensionality of the input vector), *l=3* (the dimensionality of the output vector). Matrices *A*, *B*, *C* can be written as:

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \dfrac{ML_0 g b_1}{L_0\left(J_1 + b_1^2 m\right)} & \dfrac{-k_1}{(J_1 + b_1^2 m)} & 0 & \dfrac{k_3 b_1}{L_0 (J_1 + b_1^2 m)} \\ 0 & 0 & 0 & 0 & \dfrac{-k_2}{\left(J_2 + Mb_2^2\right)} & 0 \\ 0 & 0 & \dfrac{-g\left(J_1 + b_1^2 (m+M)\right)}{L_0\left(J_1 + b_1^2 m\right)} & \dfrac{k_1 b_1}{L_0 (J_1 + b_1^2 m)} & 0 & \dfrac{-k_3\left(J_1 + b_1^2 (m+M)\right)}{ML_0^2\left(J_1 + b_1^2 m\right)} \end{pmatrix}, \tag{5.16}$$

$$B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \dfrac{1}{J_1 + b_1^2 m} & 0 \\ 0 & \dfrac{1}{J_2 + Mb_2^2} \\ \dfrac{-b_1}{L_0\left(J_1 + b_1^2 m\right)} & 0 \end{pmatrix}, C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

The initial conditions vector x_0 is defined in (5.8).

5.5.2. Analysis

In this section we will considere the main physical factors, which are taken into account in various GC models, to perform the qualitative and the quantitative analyses of the GC reference model and to facilitate the interpretation of the simulation results.

Equations (5.6) without the effect of the viscous friction have been obtained in [SS1982]. It has been found that the moments of inertia of the trolley and the hoist drives J_1, J_2 have to be taken into account in the GC model. The moments of inertia significantly affect the dynamics of the system because their values are comparable with $(m+M)b_1^2$ and Mb_2^2. The small equivalent radius of the drums b_1, b_2, which correspond to the large reduction ratio, reduces the influence of the trolley and the load on the drive motors.

In [AT1987] the authors derived the GC model with neither the viscous friction nor drives' moments of inertia. Still, the influence of the motor speed-torque characteristic on the system dynamics was analyzed. The following two cases were discussed:

- The drive motor with the velocity feedback is used. As a consequence, the load oscillations do not affect the motion of the trolley and the length of the rope because of the control system, which compensates for the load oscillations influence. To study the dynamics of the system we have either to augment the equations (5.6) with the dynamic equations of the control system or, if the control system is good enough, to describe the load-rope assembly as a physical pendulum neglecting the coupling between the load and the trolley. In the latter case the inputs are simply defined by the pre-computed velocity and acceleration profiles.
- The motor without a control system is used. In general, the motor is described by its nonlinear speed-torque characteristic (rotation velocity vs. torque). The torques T_1, T_2 in (5.6) become the nonlinear functions of the drive motor rotation velocities $\dot{\theta}_1, \dot{\theta}_2$. In [AT1987] it has been shown that for most motors the real nonlinear speed-torque characteristic can be approximated to avoid the torque being dependent on the rotation velocity. The approximation means that drive torques are set to be equal to their nominal values if the rotation velocities less than the maximum ones given in Table 5.1; the torques are set to zero if the maximum velocities are exceeded. However, if the speed-torque characteristic is approximated, the additional damping, which is due to the inclined characteristic, is neglected [Uns1976, EME1992]. The effect of that damping, though, may be even greater than the damping, which is due to the structure of the GC.

It is the second case that has been considered in this dissertation. The torques T_1, T_2 are assumed to be independent on the rotation velocities $\dot{\theta}_1, \dot{\theta}_2$ if the constraints in Table 5.1 are satisfied. The result of the total damping is taken into account as a part of the viscous friction coefficients[5] $k_i, i = \overline{1,2}$. Although there are no difficulties with modeling the nonlinear speed-torque characteristic of the drive motors in the DVP systems, in the GC reference model we have to adopt the above-described procedure because [SS1982] does not provide the information on the mechanical characteristics of the drive motors.

---

[5] Both damping, which is due to the speed-torque characteristics, and viscous friction are proportional to the rotation velocity with the physical consequence of energy dissipation. Consider equation (5.6). The larger the velocity is, the more the drive torque is reduced. The right-hand sides of the first and the second equations decrease or, reversely, the left-hand sides of those equations increase. Hence, damping can be taken into account in the same way as viscous friction

The GCs being very lightly damped, damping is only 1-5% of the critical [ANM2003]. Small damping results in the utterly long and oscillatory transient process. If a GC model is intended for the control system design only, damping can be neglected. This is the case in [SS1982], where the GC model is unstable. However, damping is important for the stability of the numerical simulation procedures when a GC model has to be simulated in an open loop [Chu2003]. Indeed, if a model has no damping and it is only marginally stable, the simulation results will strongly depend on a small variation of the initial conditions or small parametric uncertainties. Hence, the simulated behavior of the system may turn out to be either stable or unstable. Furthermore, the longer the simulation period of an unstable model is, the stronger the divergence between the simulated and the "true" trajectories is. In this chapter we will compare the dynamic behavior of the GC reference model and the GC DVP. So, the instability of the model, as in [SS1982], is undesirable[6] since the GC reference model has to be simulated without control. By introducing some damping (viscous friction) to all three DOFs of the GC reference model, the first DOF (the trolley) is made marginally (non-asymptotically) stable, and the third DOF (the rope-load assembly) is made asymptotically stable[7].

There is no information on the damping (viscous friction) of the GC in [SS1982]. Besides, this information cannot be obtained from a real crane as described in [Uns1976]. The only solution is to choose the viscous friction coefficients based on [ANM2003], see Appendix 1.

Coulomb friction has been considered in some GC models [ANM2003]. Nevertheless, it can be safely neglected in the GC reference model for the reason that the effect of Coulomb friction on the dynamics of the GC is several times less than other factors, for example, the trolley acceleration and deceleration [Uns1976].

There are GC models that describe the three-dimensional (non-planar) oscillations of the load [ANM2003]. One important result is that if the out-of-plane oscillations do occur, they may have almost the same amplitude as the in-plane oscillations. Also, it has been pointed out that the nonlinear effects become prominent when the amplitude of the oscillations is only several degrees. In [TK1998, MN2003] the double-pendulum GC model has been introduced. The model accounts for the fact that the load center of mass does not coincide with the point, to which the load is attached to the hook/spreader. Basically, this

---

[6] Another aspect of the undesired marginal stability and instability is that the identification methods seem to perform better if the system is stable. See Chapter 4 for more detail

[7] As a matter of fact, even if k_3 is zero, the third DOF will be asymptotically stable because of the nonzero k_1

would lead to extending the GC model by one additional mechanical DOF. Moreover, if one considers the fact that the mass of the load may be comparable with the mass of the crane, it would further complicate the model. For the sake of clarity, neither the non-planar load oscillations nor the double-pendulum load suspension are discussed in this dissertation.

The dynamic coupling between the trolley and the load is essential in our example. Indeed, the mass of the trolley is several times less than the maximum mass of the load: $m = 6 \cdot 10^3 kg; M = 4{,}25 \cdot 10^4 kg$. When the hoist drive machinery is located on the trolley [KCI2001], the coupling is weaker because the trolley is quite heavy ($m = 5{,}6 \cdot 10^4 kg; M_{\max} = 7{,}9 \cdot 10^4 kg$). So, there are GC models that ignore the coupling altogether [Ver2002].

The rigidity of the GC construction plays a vital role in insuring the dynamic quality of the crane operations. If the GC construction is not rigid enough, it is virtually impossible to suppress the load oscillations [Rob2003]. The dynamic effect of the crane rigidity has been studied in [OHH1999], where the GC was modeled as an elastic beam with a moving trolley. It has been shown that the amplitude of the beam oscillations rise as the load mass increases. If the velocity of trolley is small, the largest deflection is observed in the middle of the beam; if the fast moving trolley abruptly stops near the ends of the beam, the maximum oscillations are found at the ends of the beam. In this dissertation, the crane construction is assumed to be absolutely rigid because DVP system IGRIP allows to model only rigid structures.

In [Uns1976] there are some recommendations on modeling the torsional elasticity of the GC drives. Yet, these effects do not strongly influence the load oscillations on most cranes due to the high rigidity of their drives.

The GCs are very sensitive to the dynamic excitations at the frequency which is close to the natural frequency of crane's construction [ANM2003]. Depending on the construction and the external factors, the load oscillations can be periodic with one or several modes, aperiodic or chaotic. Thus, the nonlinear oscillations of the marine cranes have been studied in [EKM2002, EK2003]. Specifically, the *path following* method, which is used to analyze nonlinear oscillations and bifurcations, appears to be suitable for implementation on the basis of DVP, see Section 4.1.

Finally, the wind excitation can be considered either as a damping factor (the air drag) or as an external disturbance. The method for calculating the wind load is given in [SFS1977a[8], Ver2002].

---

[8] This is a Finnish national standard. The equivalent international standard is ISO 4302-1981 "Cranes - Wind Load Assessment", International Organization for Standardization, 1st ed.

In the first case, the trolley velocity is not high. Hence, the air drag is not large either. According to [SFS1977a], the wind force F, [N], which affects a body, can be defined as: $F = \frac{c}{1.6} \cdot v^2 \cdot S$, where v is the wind speed (equivalently, the velocity of a body in the static atmosphere), [m/s]; S is the area of the front side of the body, [m^2]; c is the aerodynamic coefficient that depends on the on the shape of the body. For a standard ISO 1AA-type container, which is 40 feet (12.0 m) long and 8 feet 6 inch (2.6 m) high [ISO1995], $S = 31.1 m^2$ and $c = 1.3$. The maximum trolley velocity being $v_{1\max} = 2.5\, m/s$, the air drag is only $F = 0.16\, kN$.

In the second case, the wind force may be significant. The wind pressure for a working crane is assumed to be $250\, Pa$ [SFS1977a]. The ***side*** wind force, which acts on an ISO 1AA container, is just $1.9\, kN$, whereas the ***front*** wind force is $10.1\, kN$. Consequently, the front wind force results in the load displacement that is comparable with the displacement, which is due to the trolley acceleration and deceleration.

Let us note an interesting detail regarding the GC construction. The containers are known to be carried in the sea-land direction along their short side. There are two reasons for doing so. First, container's moment of inertia is less along the short side than along its long one. This fact allows to reduce the oscillateness of the system. Second, in this configuration, the side wind force, which is more difficult to tackle than the front wind force, is smaller than the front wind force. The side wind force is opposed by the large moment of inertial along the long side of the container. The front wind force, in its turn, is compensated in the same way as the load oscillations, which are excited by the motion of the trolley.

In our example, the wind excitation could have been easily modeled in DVP IGRIP either as an external force or as viscous friction.

Having discussed the consistency and the adequacy of the GC reference model, we will now consider the aspects directly related to the control system design. First of all, we have to analyze the stability of the GC reference model.

The linearized system (5.16) is marginally stable. The characteristic polynomials of the trolley DOF (no term with $\dot{x}_6$) and the hoist DOF lack the terms with the zero derivative of the outputs. Both polynomials have zero roots. Therefore, system matrix A has two zero

---

(1981), and Russian national standard GOST 1451-77 "Krany gruzopod'emnye. Nagruzka vetrovaya. Normy i metod opredeleniya" (1977) eigenvalues[9]. As a result, the first (indirect) Lyapunov's method [Zak2003, AKN1998] cannot be used to analyze the stability of nonlinear system (5.6). Either the second (direct) Lyapunov's method (stability conditions) [Zak2003] or Chetaev's theorem (instability conditions) [AKN1998] needs to be employed. In our example the rough qualitative analysis of the system stability is sufficient. In addition, the stability of the GC can be assessed by means of a direct simulation.

Thus, if the rails are even and horizontal, then the trolley DOF in (5.6) is non-asymptotically stable, thanks to the energy dissipating viscous friction. The bending deflection of the rails and the nonlinear static friction might further stabilize the trolley. Again, the viscous friction makes the load angular displacement DOF asymptotically stable. The hoist DOF in (5.6) is unstable even with the gravity compensating the torque Mgb_2 in (5.10). In practice, such a linear compensation is not capable of compensating for the nonlinear effect of the gravity $-Mgb_2 \cos\varphi$ in (5.6). Term $-Mgb_2 \cos\varphi$ depending on the load angular displacement, the rope length will either increase or decrease as the load oscillates. If the rope length is not fixed, it is the hoist drive that unstabilizes the nonlinear system (5.6). This conclusion has been proved by the simulation results[10].

One of the key applicability conditions of the DVP control system development method is the plant stability, see Sections 3.1.4 and 4.1. The significance of the plant stability is explained by the need to keep the phase trajectories of the open-loop plant in the neighborhood of the work region to obtain the simulation data, which describes the plant in this region. In addition, it is difficult to simulate unstable pants over long time intervals [Chu2003]. However, the considered class of plants can be extended to include those unstable plants, which do not leave the work region of the phase space throughout the simulation process. Summing up, the proposed control system development method is applicable to the GCs with the following constraints:

- The GC has two DOFs. The rope length is fixed by locking the brakes. The second equation is removed from (5.6) and (5.16). The nonlinear system is non-asymptotically stable;
- The GC with three DOFs. The system is somehow retained in the vicinity of the work region. This condition can be proved by means of the direct simulation.

---

[9] More formally, it can be shown that here the modes $(a_i + b_i t)e^{\lambda t}, i = \overline{1,6}$, which correspond to the repeated eigenvalue at zero $\lambda = 0$, are bounded because $b_i = 0, \forall i$

[10] This corresponds to Lagrange's stability principle, that is, an equilibrium is stable if and only if the value of the potential energy is a relative minimum. Obviously, the minimum potential energy of the GC can be achieved when the load rests on the ground

In both cases, the input signals, which stimulate the system to return to the neighborhood of the work region, may have a positive impact on the identification results[11].

Concluding this section, we note that the linearized system (5.16) is fully observable and controllable because the Kalman's observability and controllability matrices have full rank [Zak2003]. This can be checked with functions `obsv` (observability), `ctrb` (controllability) and `rank` of MATLAB / Control System Toolbox [MLAB2005].

5.5.3. Simulation

The purpose of this section is to simulate the nonlinear GC model (5.6) and the linearized GC model (5.16) in a typical work regime and to assess the quality of the linearization. MATLAB has been chosen to be the simulation engine.

Assume that initially the GC is at rest. Let us apply the identical input signals to both models:

- Torque T_1: accelerating for $t = 2\ s$, $T_1 = 0.5T_{1\max} = 515\ N \cdot m$ (it is well within the constraints in Table 5.1); traveling with constant velocity[12] for $t = 4\ s$, $T_1 = 0$; decelerating for $t = 2\ s$ (the trolley is not to be fixed by brakes), $T_1 = -0.5T_{1\max}$. The amplitude and the duration of the applied torque are such that the constraints in Table 5.1 are satisfied. The initial position of the trolley is 10 m right of the leftmost end of the rails, i.e. $\theta_1(0) = 694.44\ rad$;
- Torque T_2: hoisting with acceleration for $t = 2\ s$; after that hoisting is performed with constant velocity[13]. Torques T_1 and T_2 are applied simultaneously. Initial length of the rope $L_0 = 28\ m$, i.e. $\theta_2(0) = 2295.1\ rad$. The amplitude and the duration of the applied torque are such that the constraints given in Table 5.1 are satisfied. Thus, $T_2 = -Mgb_2 + T_{\max}$, where the first term is the load balancing torque and the second term is the relative effective torque ($T_{\max} \le 0$ for hoisting). To satisfy the constraints, the following inequality has to hold:

$$\left|T_{\max}\right| \le \left(Mb_2 + {J_2}/{b_2} \right) \frac{V_{2\max}}{t_{acceler}}. \tag{5.17}$$

---

[11] E.g. a zero-mean signal

[12] We mean the programmed ideal velocity profile, which is not influenced by the load oscillations and the varying rope length

[13] See the preceding footnote

The inequality (5.17) directly follows from the second equation in (5.11) when friction is neglected. It is important to take into account the influence of the trolley and the load oscillations on the instantaneous hoisting velocity. Hence, the actual $|T_{\max}|$ has to be less than the maximum allowed torque. The fact that the constraints have been indeed satisfied has to be checked by means of the direct simulation.

The linearized system (5.16) has been simulated with the use of function `lsim`, which is included in MATLAB / Control System Toolbox [MLAB2005].

The numerical integration of the nonlinear equations (5.6) has been accomplished by employing MATLAB function `ode45`, which implements the Runge-Kutta method of (precision) order 4-5 with the adapting step size [MLAB2005, WT1997, KNM1989, BPV1966]. The relative tolerance has been set to be 10^{-6}. The first derivatives of the state variables (Cauchy form) have to be evaluated at each step. Since $\theta_1, \theta_2, \varphi, \dot{\theta}_1, \dot{\theta}_2, \dot{\varphi}, T_1, T_2$ are known, we only have to obtain $\ddot{\theta}_1, \ddot{\theta}_2, \ddot{\varphi}$. Note that equations (5.6) are linear with respect to these variables. Hence, we can write equations (5.6) as:

$$A_* \begin{pmatrix} \ddot{\theta}_1 & \ddot{\theta}_2 & \ddot{\varphi} \end{pmatrix}^T = B_* . \tag{5.18}$$

At each step, the elements of the matrices A_*, B_* have to be evaluated and the linear system of equations (5.18) has to be solved. The method of Gauss elimination has been utilized (operator "back slash" (\) in MATLAB) [KMN1989, MLAB2005]. The condition number is $cond(A_*) \approx 10^6$ (floating point relative accuracy is about 10^{-16}), although the norm of the residuals does not exceed 10^{-12}, which is much less than the practical precision of the elements in A_*, B_*.

Generally, when the condition number $cond(A_*)$ is very large, the problem of solving linear systems is badly conditioned and the uncertainty of A_*, B_* and the roundoff errors strongly influence the solution. To tackle the problem, the regularization techniques [Tik1995], scaling [GV1989], and the singular analysis [KNM1989] can be employed.

If the equations of motion were nonlinear with respect to the derivatives of the state variables[14], then the nonlinear equations would have to be solved at each step of the numerical integration to obtain the Cauchy form. Solving the nonlinear equations, for instance with MATLAB function `fsolve` [MLAB2005], is more laborious than solving the linear equations. In our example, integrating and solving the linear equations (with matrices A, B)

[14] For example, when the plant has non-holonomic constraints

has been faster by several tens of times compared to integrating and solving the nonlinear equations (with matrices A_*, B_*), even when the initial approximation of the solution was made use of. If solving the nonlinear equations cannot be avoided, the Runge-Kutta method of order 4-5 becomes too expensive because it requires evaluation of the nonlinear function at six points. The Adams-Bashforth-Moulton methods of order 1-12, which are implemented in MATLAB as `ode113` [MLAB2005, BPV1966], may be more efficient than the Runge-Kutta method at tighter tolerances and when the function is expensive to evaluate.

If the analytical model of the plant is known, the sampling frequency can be determined in accordance with the Nyquist-Shannon-Kotelnikov sampling theorem using plant's power spectrum, amplitude-frequency characteristic, eigenvalues or transient response [Ise1977]. A difficulty of the proposed DVP control system development method is that the analytical model is unknown. Hence, the sampling frequency can be determined using some expert knowledge and/or the direct simulation. In the latter case we rely on the underlying integration procedure, which is assumed to be capable of maintaining its consistency by adapting the integration step size [PTV1999, WT1997].

The GC is a low-frequency system. The desired rise time (10% to 90% of the step response) of the closed-loop system is about *5-10 s* (see Section 5.6.4). According to [SSH1996] the sampling period has to be $(0.4 \div 0.06)T_{rise} \approx 0.3 \div 2.0\,s$. Let us set the sampling time to *0.1 s* because the corresponding Nyquist frequency $(5\,Hz)$ should well exceed the GC closed-loop crossover frequency ($\approx 0.3 \div 0.1 Hz$ for the above-defined rise time of a one-overshoot controlled step response) and, in accordance with [SSH1996], $\frac{1}{30 f_{crossover}} < T < \frac{1}{5 f_{crossover}}$ yielding $0.1s < T < 0.7\,s$. The chosen sampling frequency does not seem to be problematic from the point of view of the technical implementation of the control system.

The MATLAB program for simulating the nonlinear and the linear models of the GC are presented in Appendix 2 (`lin_vs_nonlin.m`). The simulation results are shown in Figure 5.2. It is clear that the linearized model qualitatively and quantitatively well approximates the nonlinear process in the work region. The load oscillations significantly influence the motion of the trolley. The load oscillations vanish rather quickly due to the viscous friction of the trolley drive (k_1). It can be seen that the phase variables remain within the work region and that the constraints in Table 5.1 are met. In [Uns1976, SFS1977a] there are some estimates of the maximum load angular displacement and the oscillations period as functions of the applied torques. The simulation results are consistent with the estimates.

Generally, such estimates along with any other additional information, including expert knowledge, assist in validating the simulation results, which are obtained with analytical models or DVPs.

It is convenient to analyze the load oscillations in the phase plane (see the bottom plot in Figure 5.2) since the regimes of accelerating, moving with constant velocity, decelerating, and load oscillations decaying (marked with numbers) are clearly visible. An interesting feature of the plant is that decreasing the rope length leads to increasing the load oscillations amplitude. This is a typical nonlinear effect, which is explained by the energy conservation law. It is not observable on the linear GC models and it is especially prominent when viscous friction is small. This effect is used in practice by the GC operators [Uns1976].

5.6. Modeling and control system design of the gantry crane

Henceforth we assume that the analytical model of the GC is not available. The development of the GC control system will be accomplished exclusively on the basis of the GC DVP.

In practice, the GC DVP may already exist. Should it be the case, the control system developer can directly proceed to the identification and the control system design stages. However, the designers of the GC may provide the control system developer with a VP, which is lacking the information on the dynamic properties of the plant. So, such a VP has to be extended to include the necessary dynamics information. Finally, if neither the DVP nor the VP of the GC is available, it is necessary to construct the DVP from scratch.

In the following sections it will be considered the application of constructing the DVP, conducting the identification experiments, and designing the control system to the GC.

The input of the DVP development method is the structure and the parameters of the GC reference model (see Figure 5.1 and Table 5.1) along with some qualitative information on the stability of the plant and the interconnections between its DOFs. The output of the method is the GC control system (the control law), which is obtained with neither the analytical model nor the physical embodiment of the GC.

To prove the adequacy and the quality of the results, the GC DVP simulation and the identified linearized model are compared with the results acquired with the corresponding GC analytical model.

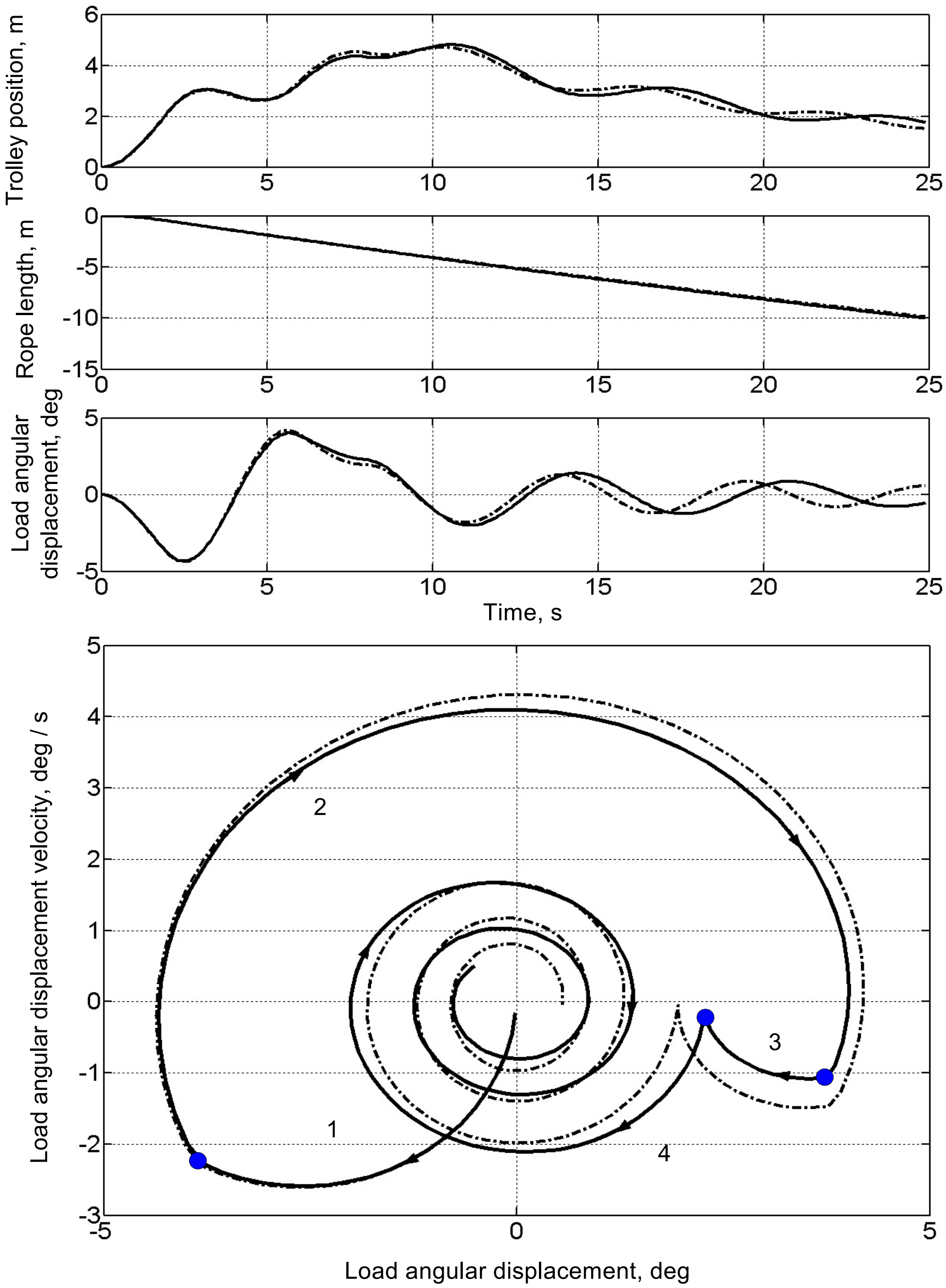

Figure 5.2. Simulation results of the GC models in a typical work regime (solid line: the linearized model; dash-dotted line: the nonlinear model). At the bottom is the phase portrait of the GC for the load angular displacement coordinate. The programmed trolley motion: 1 – accelerating; 2 – moving with constant velocity; 3 – decelerating; 4 – being at rest

5.6.1. DVP construction

The purpose of the section is to construct the GC DVP, which is suitable for the subsequent analytical model identification and control system design.

Note that if the lifecycle of the GC is supported by a (D)VP/CALS system, then the DVP, which is needed for the control system development, may be already constructed at the stages preceding the control system development. In this ideal case the control system developer does not have to construct the DVP from scratch and, of course, the material of this subsection would be unnecessary.

In order to construct the DVP, which would be suitable for simulating the GC dynamics, the structure (Figure 5.1) and the parameters (Table 5.1) of the GC reference model have to be defined in a DVP system. In our example care has to be taken of keeping the equivalence of the DVP and the analytical model to enable the comparison between the results. From the point of view of visualization, the level of details of the DVP can be quite low. In other words it is sufficient to construct a simple wire model of scale 1:1 with the actual geometry (the size and the configuration of the main construction elements), kinematics (the kinematics graph), and dynamics (mass, friction etc.) properties.

In sum, the GC DVP construction comprises the following three principal stages: defining geometry (1), defining kinematics (2), and defining dynamics (3) of the GC. In our example we have used DVP/CALS system IGRIP.

1. ***Defining the geometry*** of the GC DVP, Figure 5.3. In [SS1982] there is no reference to the dimensions of the drives, girder and trolley. Since the physical parameters such as mass, moments of inertia, friction can be set ***explicitly*** in IGRIP, the above-mentioned dimensions can be arbitrary because they do not affect the dynamics of the system. Indeed, the girder does not move, the motion of the trolley is translational, and the total moments of inertia of the drives can be set explicitly while defining the dynamics of the plant. In brief, the motors, the drums, and the reduction gears of the trolley and the hoist drives are modeled as cylinders with arbitrary radius (e.g. 0.5 *m*) and height (e.g. 0.5 *m*). The girder is represented as a cuboid with arbitrary width (e.g. 4 *m*), height (e.g. 0.1 *m*), and length (e.g. 40 *m*). The trolley is modeled as a cuboid with arbitrary size (e.g. width: 4 *m*; length: 2 *m*; height 1 *m*).

If the DVP was constructed in accordance with the actual technical drawings and specifications, the above-mentioned dimensions would not be arbitrary. On the contrary, using the true geometric properties of the GC elements, the VP system would be able to calculate, for example, their moments of inertia.

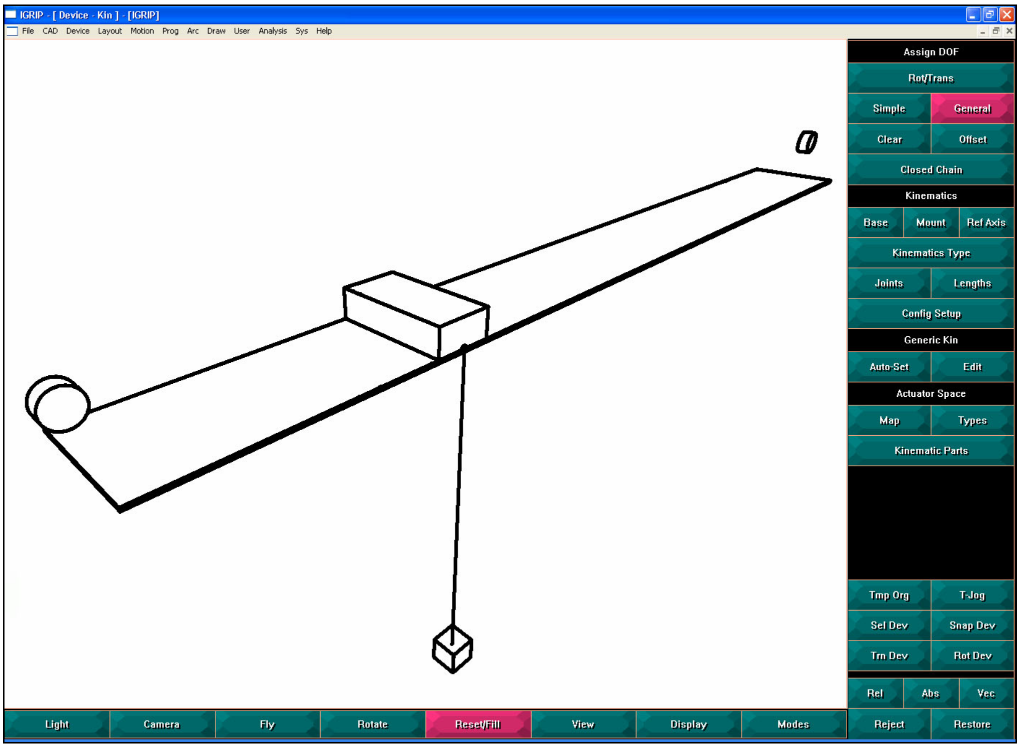

Figure 5.3. Geometry of the GC DVP as it appears in the window of the DVP system IGRIP

The situation with the load is different. According to the GC reference model, the load is modeled as a material point. However, all dynamic elements in the DVP have to have finite dimensions and non-zero moments of inertia. Since the load revolves around the axis that does not pass load's center of mass, its non-zero moment of inertia will affect the dynamics of the system[15]. It would be more natural to specify the standard dimensions of the ISO container and to use the load mass from the reference model in order to automatically compute load's moment of inertia in a DVP system. But in our example the equivalence between the DVP and the analytical model has to be preserved. Therefore, the load dimensions should be so small that the moment of inertia about its center of mass could be made negligible in comparison with the load mass multiplied by the square of the rope length. Thus, let the load be modeled as a cube with its side of 0.1 *m* in length resulting in the moment of inertia of 71 $kg{\cdot}m^2$ ($71 \ll 42500{\cdot}28^2 = 3.3{\cdot}10^7$ [$kg{\cdot}m^2$]). For the purpose of visualization, let us represent the load as a cube with its side of 1 *m* in length, but set its moment of inertia to 71 $kg{\cdot}m^2$.

2. ***Defining the kinematics*** of the GC DVP. The kinematic graph of the GC reference model is shown in Figure 5.4.

[15] Due to the parallel-axis theorem (J. Steiner), the load's moment of inertia about the suspension point at the trolley equals the load's moment of inertia about its center of mass plus the mass of the load multiplied by the square of the rope length [Tsa1999]

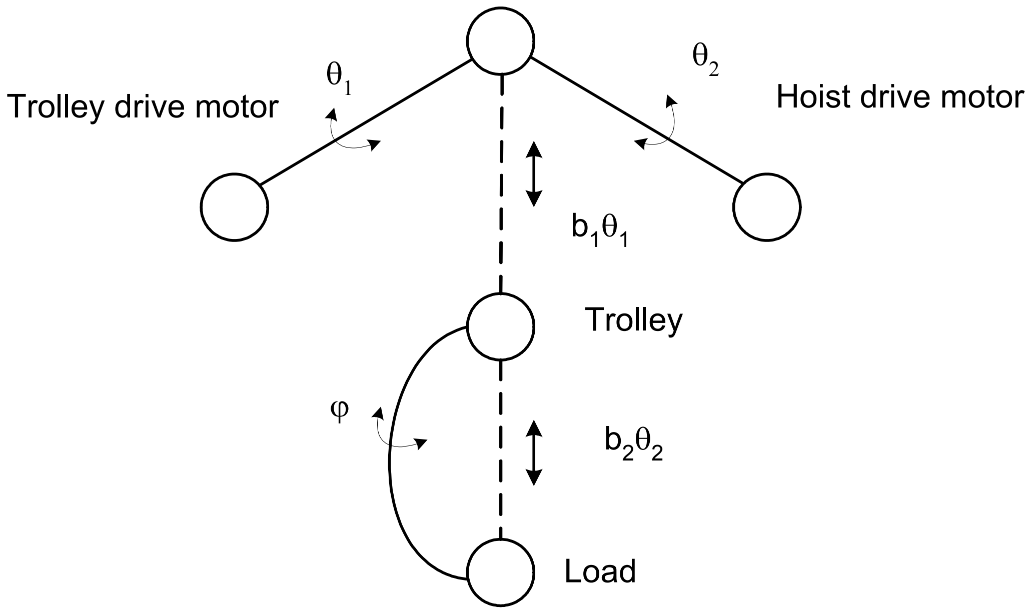

Figure 5.4. Kinematic graph of the GC reference model

Let the angular coordinates of the trolley and hoist drives be the ***driving*** coordinates, and the linear coordinates of the trolley and the rope length be the ***driven*** ones. As a result, the kinematics graph will have:

- ***Five nodes***, which correspond to the kinematical links: the girder, the two motors, the trolley and the load;
- ***Three driving edges***, which correspond to the following three DOFs: the trolley and the host drives, the load angular displacement, and ***two driven edges***: the linear coordinates of the trolley and the load. There are five edges (joints) altogether, but there are only three DOFs.

The functional relationship between the driving and the driven joints can be defined in the properties of the driven joints as[16]: $dof(1)\cdot 1{,}44\cdot 10^{-2}$ for the trolley and $dof(2)\cdot 1{,}22\cdot 10^{-2}$ for the rope length.

3. ***Defining the dynamics*** of the GC DVP. The drives' moments of inertia (J_1 and J_2, Table 5.1), the mass of the trolley and the load (m and M, Table 5.1), the load's (main) moment of inertia (71 $kg{\cdot}m^2$, see above) have to be defined in the DVP system. Finally, the friction $k_1 = k_2 = 0.92\, kg\cdot m^2\cdot s^{-1}$, $k_3 = 3.9\cdot 10^5\, kg\cdot m^2\cdot s^{-1}$ and the sampling period $T = 0.1s$ also have to be set.

[16] GSL IGRIP syntax is `dof(1)*1.44e-2*1000` and `dof(2)*1.22e-2*1000`. The units of variables `dof()` (degree-of-freedom) are ***radians***. Since the linear variables in IGRIP are measured in ***millimeters***, scaling coefficient "`1000`" is needed to convert ***meters*** into ***millimeters***

5.6.2. Identification experiment

The goal of the identification experiment, which is conducted in the DVP system, is to obtain the data-based model of the plant, which is needed for the subsequent analytical model identification and control system design.

Neither the analytical model of the plant, nor its physical embodiment will be required. Simulation in the DVP system will be the basis of the identification experiment. The proper input signals (T_1 and T_2 in our example) have to be defined for this purpose. There are a number of requirements to the input signals used for identification (see Section 4.2.1.1). Thus, the persistent excitation order should be sufficiently high; the spectrum of the input signal has to be concentrated within the bandwidth of the (open-loop) plant and about the crossover frequency of the (closed-loop) plant; if the plant is unstable with the slowly diverging trajectories, the input signal has to stimulate the plant to remain within the desired work region.

In accordance with the formulated GC control problem (see Section 5.2), the work region of the GC is defined as a neighborhood of the load fine positioning terminal point. The GC is invariant to the trolley linear position. Therefore, the range of the neighborhood is determined by a small variation of the rope length and the load angular displacement[17]. Referring to the literature [Uns1976, SS1982] and to the simulation results (see Section 5.5.3), we assume the variation of the rope is a few meters about the nominal rope length of 28 *m* and the amplitude of the load oscillations is not more that 10°. The maximum velocities, accelerations and torques are to be determined in accordance with Table 5.1.

Let us define the input signals for the identification and the validation experiments[18]. The programmed motion of the trolley (the input torque T_1) comprises the flowing phases (the information for the validation experiment is given in brackets):

- Accelerating for $t = 1s \quad (t = 1.5\,s)$;
- Moving with constant velocity for $t = 3s \quad (t = 4\,s)$;
- Decelerating for $t = 1s \quad (t = 1.5\,s)$;
- Being at rest for $t = 8s \quad (t = 6.5\,s)$.

The wave form of T_1 is a meander with amplitude $0.5T_{1\max} = 513\,N \cdot m \quad (0.6T_{1\max} = 618\,N \cdot m)$. The amplitude of the signal is such that the constraints on the phase coordinates of the plant

---

[17] This is equivalent to the conditions of quasistationarity and model linearity

[18] See Appendix 2 (function `input_gen.m`) for more detail

are satisfied. The total duration of each cycle is $t_{total} = 13\,s\ (t_{total} = 13.5\,s)$. There are three trolley motion cycles to the left and to the right, i.e. six cycles altogether. The zero-mean Gaussian white noise is added to the input signal T_1 to increase its persistence of excitation. The standard deviation of the noise is 0.1 of the maximum amplitude of the meander.

Theoretically, if the input signal (without noise) is not sufficiently persistently exciting, adding some white noise would make the persistent excitation order of the signal infinite. Despite this fact, the identification results obtained with such a signal will not be reliable enough. The reason is that white noise has a uniform power spectrum. Hence, it is not suitable for exciting the low frequency modes of the plant, which are usually crucial for identifying an accurate dynamic model. In other words, the spectrum of the input signal has to be focused within the bandwidth of the plant and about the crossover frequency. Nevertheless, we have experimentally found out that adding some noise does have a positive impact on the identification results owing to the improved condition number of the identification design matrices (see Section 4.2.1.1).

The programmed motion of the load, the input torque T_2, contains four cycles of hoisting and lowering. Each phase of the cycles (accelerating, moving with constant velocity, decelerating, being at rest) is executed for $t = 2\,s$. The input signal has the gravity balancing bias $-Mgb_2$, which is a necessary condition for linearizing the GC model. The amplitude of the meander is equal to the maximum torque, which satisfies the constraints in Table 5.1 and formula (5.17). The amplification coefficient is set to 0.8 (0.7). The zero-mean Gaussian white noise is added to T_2. The standard deviation of the noise is 0.01 of the maximum unbiased amplitude of the meander.

The total duration of the input sequence is $81\,s\ (85\,s)$. Since the sampling period is $T_s = 0.1\,s$, the sequence length is $N = 810\ (N = 850)$.

The waveforms and the energy spectra (***excluding*** the contribution of the "negative" frequencies) of the input signals T_1 and T_2, which are used in the identification and the validation experiments, are shown in Figure 5.5.

The order of the persistence of excitation for the input signals has to be computed to insure that the input signals are suitable for identification. For more detail see Section 4.2.1.1 and Appendix 2, function `input_gen.m`. Let the input matrix U have I_{max} column blocks with $I_{\max} > n$, where n is the supposed order of the model to be identified. For the persistence of excitation order of the input signal to be $2I_{\max}$, the rank of the input covariance matrix has to be $2mI_{\max}$ (full rank), where m is the dimension of the input signal (m=2 in our example).

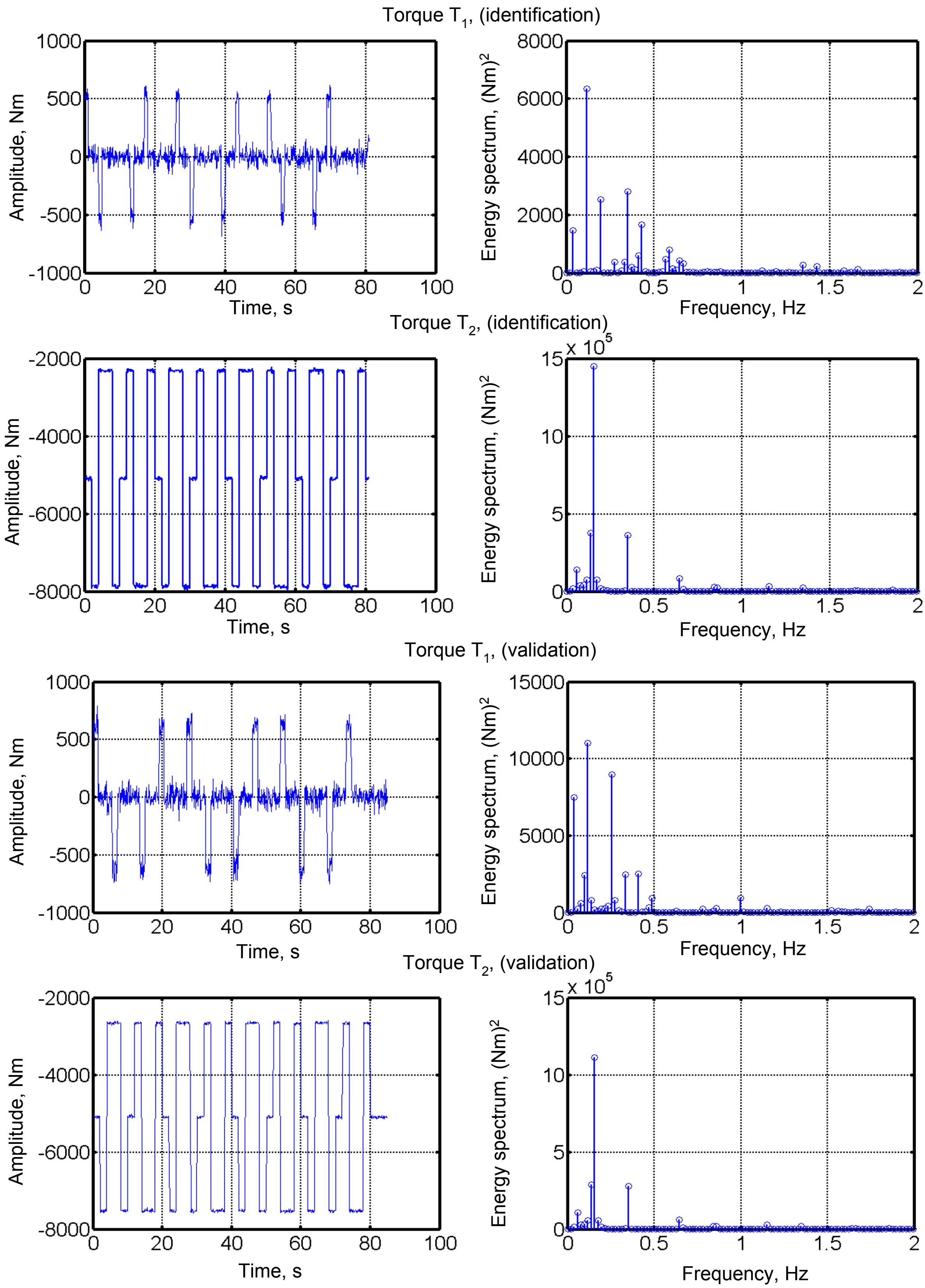

Figure 5.5. Waveforms and energy spectra of the input signals used in the identification and the validation experiments

The calculations show that for the noise-free input signal, which has been defined earlier in this section, the input covariance matrix UU^T does have full rank for the values of $I_{\max}$ up to 40. It means that the input signal is theoretically suitable for identifying a linear model of the order not exceeding 40. Adding white noise makes the persistence of excitation order infinite (though white noise, as it has been already discussed in this section, is not very efficient in practice).

Furthermore, the row dimension *j* of the matrix *U* has to be $j = N - 2I_{\max} + 1 >> 1$. The following inequalities must hold: $I_{\max} >> n$ and $N >> 1$. Therefore, the actual order *n* of the identified model should not be too large. If $n = 6; N = 810; I_{\max} = 100$, then $I_{\max} >> n$ and the inequality for j holds: $j = N - 2I_{\max} + 1 = 810 - 2 \cdot 100 + 1 = 611 >> 1$. The analogous calculations are applicable to the input signal for the validation experiment.

It is known that the spectrum of the input signal has to be concentrated within the frequency band where it is important to have an accurate dynamic model [Lju1999, LL1996]. Moreover, since the GC is basically a low-pass filter, the spectrum of the input signal has to overlap with a considerable part of the GC open loop bandwidth. It is essential for avoiding the situation when the “connection” between the input and the output is lost. The “connection” loss means that the input signal is almost entirely filtered out by the plant and it does not show up at the output. On the one hand, the input signal has to excite a low-frequency open-loop plant. On the other hand, the spectrum of the input has to contain the high-frequency component to cover the wider bandwidth of the closed-loop system. Certain balance should be maintained to overcome this contradiction. One remedy could be the identification in the closed-loop, when some preliminary control system is available (see Section 4.2.1.1).

At the qualitative level, both input ***and output*** have to have a sufficiently complex form[19] to insure the reliability of the identification results. Simulation of the GC has shown that the optimal period of the input signals is about 10-15 *s*. In this case the power spectra contain the clear low frequency peaks, which are necessary for identifying a low-pass plant. Apart from the dominant harmonics the spectra contain a number of higher frequency harmonics to excite the high-frequency modes (see Figure 5.5). In sum, the input signals appear to be capable of exciting the GC quite thoroughly.

[19] Specifically, the signals should contain many harmonics. This requirement for the input signal correlates with persistence of excitation [AW1995]

The input signals of the chosen form maintain the phase coordinates of the GC within the defined work region. Indeed, the load and the trolley are excited symmetrically with respect to the operation point.

Having discussed the general requirements to the input signals, we have to admit that usually the optimal form of the signals is to be found iteratively by means of simulation.

It is worth mentioning that the author of this dissertation has conducted an additional study of [FDV1998] and [VD1996], which has proved that the above-defined input signal would be also required for the data-based LQG method (see Section 4.2.2.1)

The simulation results obtained with IGRIP are presented in Figure 5.6. The input signals (excitation) and the output signal (dynamic response) form the data-based dynamic model of the GC. The data-based model is further employed for the consequent identification of the GC analytical linear model and the control system design. The format of the data-based (table) model is: $\{t\,[s], T_1[N\cdot m], T_2[N\cdot m], \theta_1[rad], \theta_2[rad], \varphi[rad]\}$.

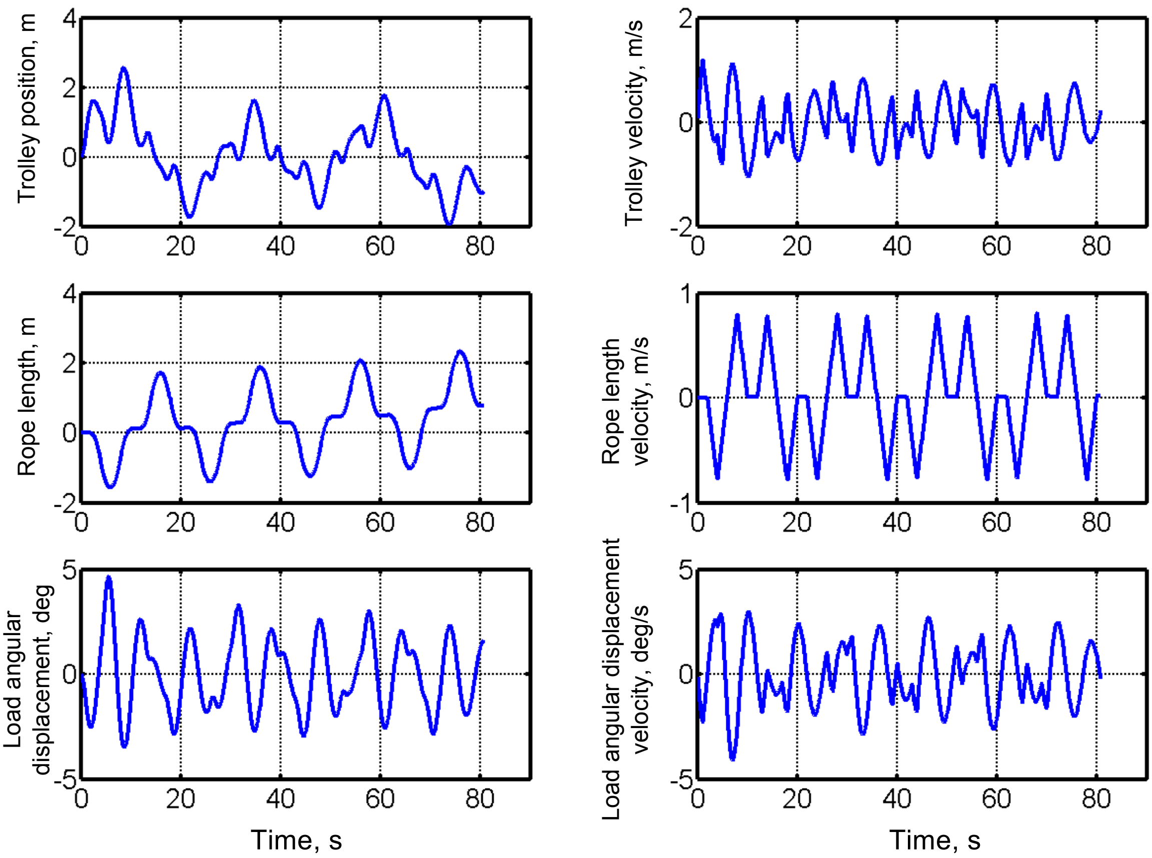

Figure 5.6. Simulation results of the GC in IGRIP

Let us study Figure 5.6. First, the phase variables do remain within the work region. Second, the oscillations of the trolley drive are multimodal thanks to the strong dependency of the trolley on the load sways and the successful choice of the input signal spectrum. Third, the oscillations of the rope length are rather simple and copy the corresponding input signal. Not only is the spectrum of the input signal a problem, but also the dynamics of the rope length does not tightly correlate with the dynamics of the trolley and the sways of the load. Fourth, the instability of the rope length DOF leads to a noticeable drift.

The comparison of the simulation, which has been obtained in IGRIP, and the simulation of the analytical nonlinear model shows that the obtained data-based model is perfectly correct, see Figure 5.7.

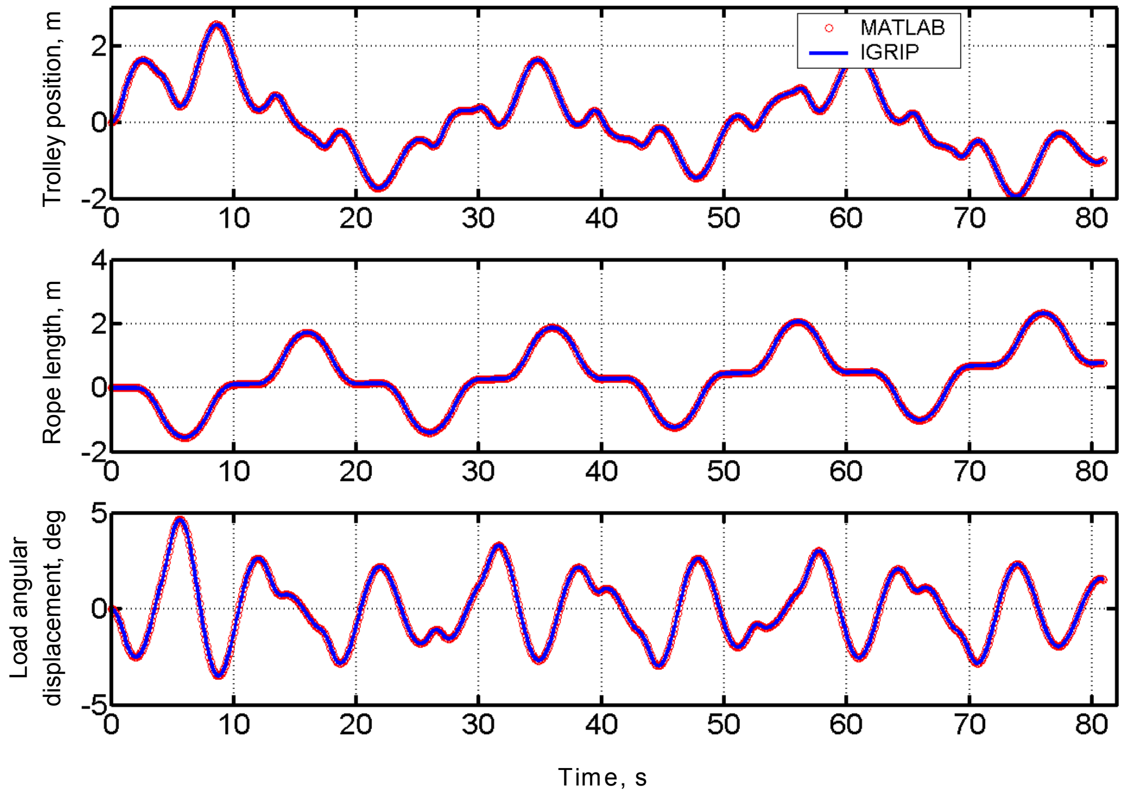

Figure 5.7. Comparison of the GC DVP and the nonlinear analytical model

Let us consider the implementation issues of the identification experiment. If the input signals are complex, the simulation program in IGRIP can make use of the external input file to read the input signal, which is generated, for example, in MATLAB. It turn, the output signal from the DVP of the simulated plant can be recorded to another external file for further processing in MATLAB. Using external files is the simplest way of communicating between the DVP systems and the CACSD systems.

The quality of the simulation results depends on the integration procedure. Usually the user can choose from a set of standard integration routines, e.g. Runge-Kutta of order 4-5, Bashforth-Moulton, predictor-corrector and so on. Furthermore, such parameters of the integration procedures as method's order, integration step size, and relative tolerance can be adjusted[20]. The consistency of the integration results can be assessed by means of comparing the results obtained with several different integration procedures.

While developing the example described in this chapter, the author of this dissertation has gained a significant experience on modeling and simulation in the DVP system IGRIP. Some of the practical issues are not reflected in the technical documentation of IGRIP. Thus, there may be problems with the video stream synchronization if the simulation is run in the real time. Hence, it may be advisable to switch off option "`real time`" and to visualize the simulation results after the simulation process has completed. The pace of the simulation process is most reliably regulated from the simulation program with function "`delay`". Declaration "`sim_update`" has to be avoided because it results in an undesired time delay, which is equal to the sampling period. In addition, some recommendations regarding the usage of innumerable IGRIP settings have been obtained. For example, the following settings have to be declared to delegate the control of the simulation pace to the simulation program: "`$DYN_TERM_CHECK = FALSE`", "`MOTION_BASIS = VELOCITY`" and "`$TIME = 0`". According to our experience, if the user (engineer) does not have sufficient practice in working with the DVP software, care has to be taken of interpreting the simulation results and every effort has to be made to validate and cross-check them.

5.6.3. Identification of the linearized model

In this section we will consider the identification of the linear analytical model of the GC using the data-based dynamic model, which has been obtained in the previous section. The subspace identification method has been chosen for identifying the GC model in the state space. The subspace method is implemented in MATLAB / System identification toolbox, function `n4sid` [MLAB2005, Lju1999, VD1996]. The program for identifying the linear model of the GC using the data-based model is presented in Appendix 2, function `crane_id.m`.

The results of the singular analysis for the model order selection in the subspace identification method are presented in Figure 5.8. Singular analysis allows us to draw the conclusion that the most suitable model order is six. This conclusion is also justified by the very structure of the GC: it has three DOFs and, therefore, it could be suitably described by a 6th order linear model.

---

[20] The integration step size can be much smaller than the sampling period. The former is defined by the stability of the integration procedure and the relative tolerance, whereas the latter is determined by the theorem of Nyquist-Shannon-Kotelnikov

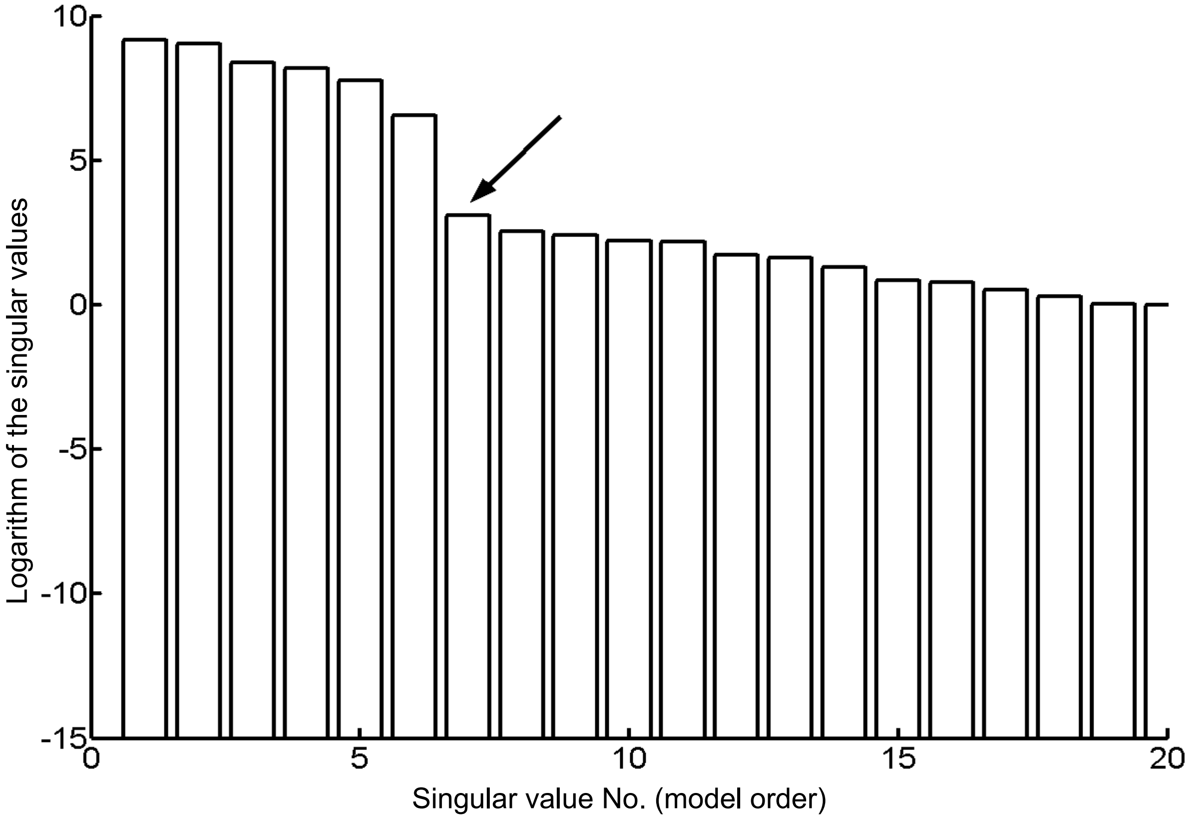

Figure 5.8. Singular analysis for the model order selection in the subspace identification method. There is an abrupt drop after the 6th singular value (marked with the arrow), which implies the 6th order of the identified model

There are a number of factors affecting the identification results. One of the most significant of them is the variable responsible for the length of the prediction horizon. The variable can be set either automatically by the identification procedure or by the user. In the former case the AIC is exploited. But we set this variable manually to {30 60 60}. It means that the `n4sid` algorithm searches for a model that would be the best (in some sense) for a 30-step-ahead prediction using 60 past outputs. This setting allowed us to achieve a good correlation between the model and the data. In general, the variable non-linearly affects the identification results and it may be difficult to find its appropriate value [Lju1996].

We will identify the state space model of the 6th order with two inputs and ***six*** outputs assuming that it is possible to measure all the state variables at the output. If we identified the GC model in the form of (5.16) with only ***three*** outputs, it would complicate our example by the need to design a Kalman or a Luenberger observer [Zak2003] to recover all six state variables from the three outputs. In fact, the assumption that the entire state vector can be measured at the output of the GC is quite reasonable because the GC state variables have a clear physical meaning (coordinates and velocities). Not only does this assumption enable to avoid the observers, but it also allows to robustify the LQ controller and to make it easier to choose the weight matrices in the LQ quality functional.

To take an advantage of the completely measurable state vector the output matrix C of the model in the state space has to be an identity matrix. Suppose we have identified matrix C' is some basis. Since C' is square (the state vector is completely measurable) and the GC is completely observable and controllable, there exists a similarity transformation T that transforms the matrix C' to $C = I$ [Zak2003]: $C = C'T^{-1}$. In our case the transformation matrix T would be simply equal to C': $C = C'C'^{-1} = I$. The transformation itself is accomplished by the MATLAB / Control system toolbox function `ss2ss` [MLAB2005].

We can measure the coordinates of the trolley, the rope length, and the angular displacement of the load directly in IGRIP, whereas it is better[21] to calculate the derivatives of these variables in MATLAB using Lagrange's interpolation polynomials of the 2[nd] order [PT1996, PTV1999]:

$$\begin{aligned} f'(t) &= \frac{1}{2T_s}\left(-3f(t) + 4f(t+T_s) - f(t+2T_s)\right) \\ f'(t+T_s) &= \frac{1}{2T_s}\left(f(t+2T_s) - f(t)\right) \\ f'(t+2T_s) &= \frac{1}{2T_s}\left(f(t) - 4f(t+T_s) + 3f(t+2T_s)\right) \end{aligned}.$$

Applying function `n4sid` and transforming the basis yield the following linear analytical model of the GC (four significant digits are retained):

$$A = \begin{pmatrix} 0.9999 & 0.0002715 & 5.887 & 0.09887 & 0.022 & -0.1316 \\ 1.673\cdot10^{-5} & 0.9995 & 0.008829 & 0.0005765 & 0.06406 & 0.7024 \\ 3.949\cdot10^{-7} & 8.596\cdot10^{-8} & 0.9965 & -4.181\cdot10^{-7} & 4.857\cdot10^{-7} & 0.09958 \\ -0.0005552 & -0.000193 & 117.1 & 0.9848 & -0.001512 & 13.28 \\ -2.795\cdot10^{-6} & 9.714\cdot10^{-7} & -0.06259 & -4.788\cdot10^{-5} & 0.9989 & -0.1014 \\ 3.282\cdot10^{-7} & 7.322\cdot10^{-8} & -0.09418 & 7.022\cdot10^{-6} & 1.328\cdot10^{-6} & 0.9885 \end{pmatrix},$$

$$B = \begin{pmatrix} 0.00117 & 8.182\cdot10^{-5} \\ 0.0002463 & 6.467\cdot10^{-5} \\ -4.444\cdot10^{-7} & 1.205\cdot10^{-8} \\ 0.01936 & -0.0003414 \\ -5.06\cdot10^{-5} & 0.00158 \\ -9.891\cdot10^{-6} & 2.292\cdot10^{-9} \end{pmatrix}, C = diag\begin{pmatrix}1 & 1 & 1 & 1 & 1 & 1\end{pmatrix}, D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

---

[21] IGRIP employs the first order polynomials, which result in unacceptable errors when the derivatives are large

The identified model qualitatively well describes the data-based model, but the quantitative congruence is relatively poor, Figure 5.9. The quality of the identified model can be also assessed by comparing the poles of the linearized model, which is derived analytically, and the poles of the identified linear model, Table 5.2. It can be seen that only 2-3 significant digits coincide, although the most considerable difference is that the identified model is asymptotically stable, whereas the analytically derived model is marginally stable.

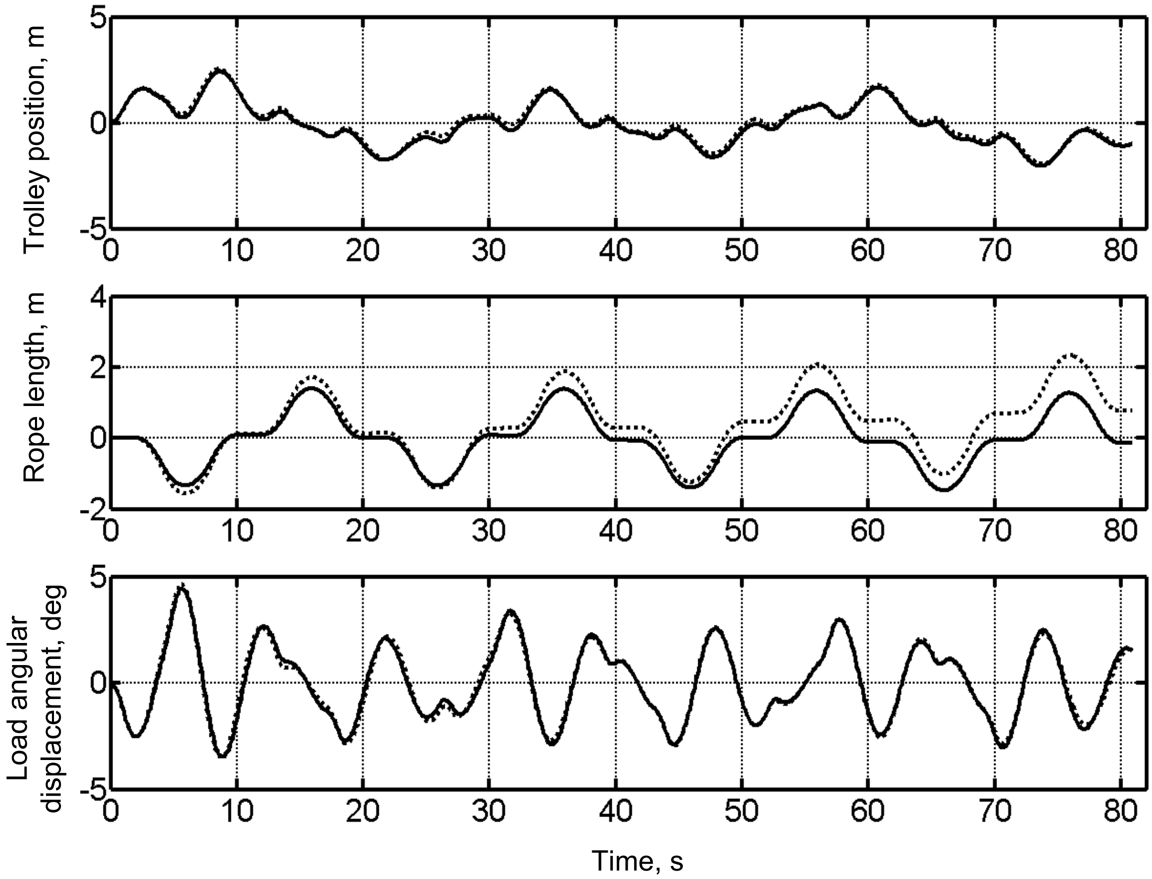

Figure 5.9. Comparison of the identified model (solid line) and the data-based model (dotted line)

Table 5.2. Poles of the identified and the analytically derived models of the GC

Poles of the analytically derived model	Poles of the identified linear model
1 0.987783 ± 0.0968388 j 0.993305 1 0.998916	0.987861 ± 0.0966951 j 0.997034 ± 0.00235532 j 0.999702 0.998640

In Figure 5.10 the singular value plots of the identified and the analytically derived models are compared[22]. Apparently the identified model accurately describes the plant around the open-loop crossover frequency. However, the description at the high and the low frequencies is relatively rough.

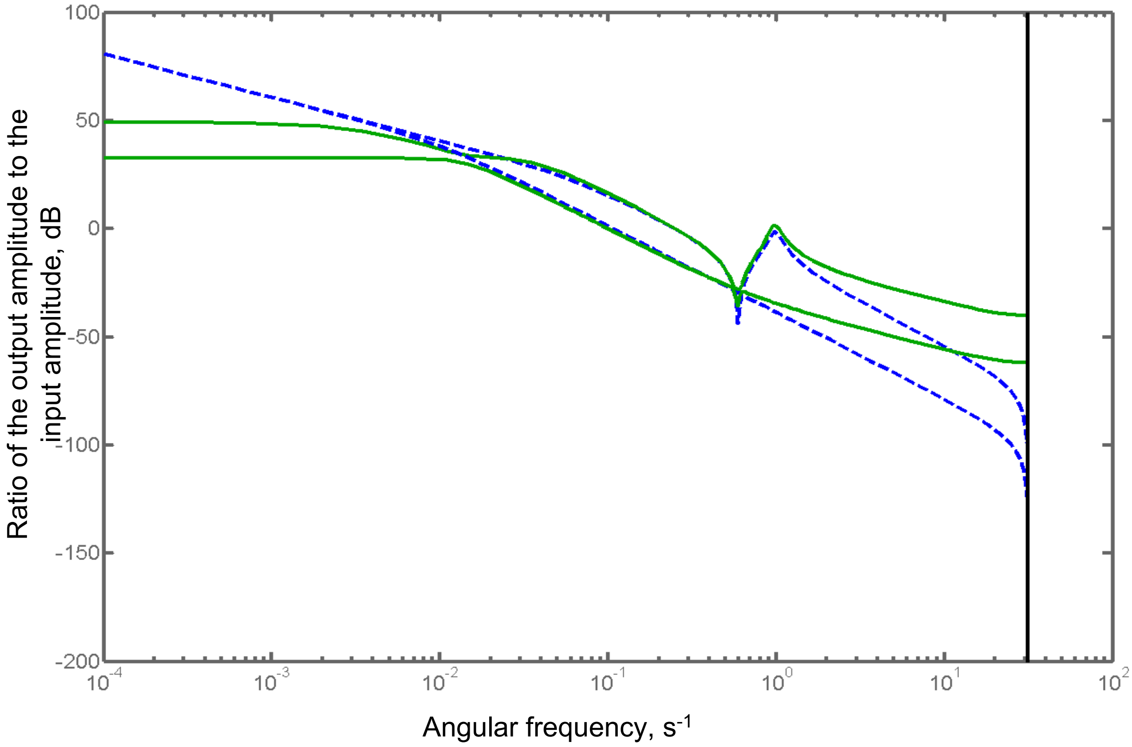

Figure 5.10. Comparison of the singular value plots of the identified model (solid line) and the analytically derived model (dashed line)

Nevertheless, the identified model describes the validation data-based model reasonably well, Figure 5.11.

The quality of the open-loop identified model – in the sense of how well it describes the data-based model – is not especially important. First, it has been already discussed that the quality of the identification results is better assessed in the closed-loop (see Section 4.2.1.1). Second, the robustness of the control system should tackle the inherent imprecision of the plant model.

The reason for the observed moderately poor quality of the identified model is that the poles of the plant are located extremely close to the unit circle (the point $z=1$) because (1) the trolley DOF is marginally stable and the rope length DOF is unstable and (2) the sampling period T=0.1 s is relatively small in comparison with the plant dynamics.

---

[22] MATLAB/Control function `sigma` [MLAB2005]; all the amplitude-frequency characteristics of the plant lay between the curves of the largest and the smallest singular values [SP1996]

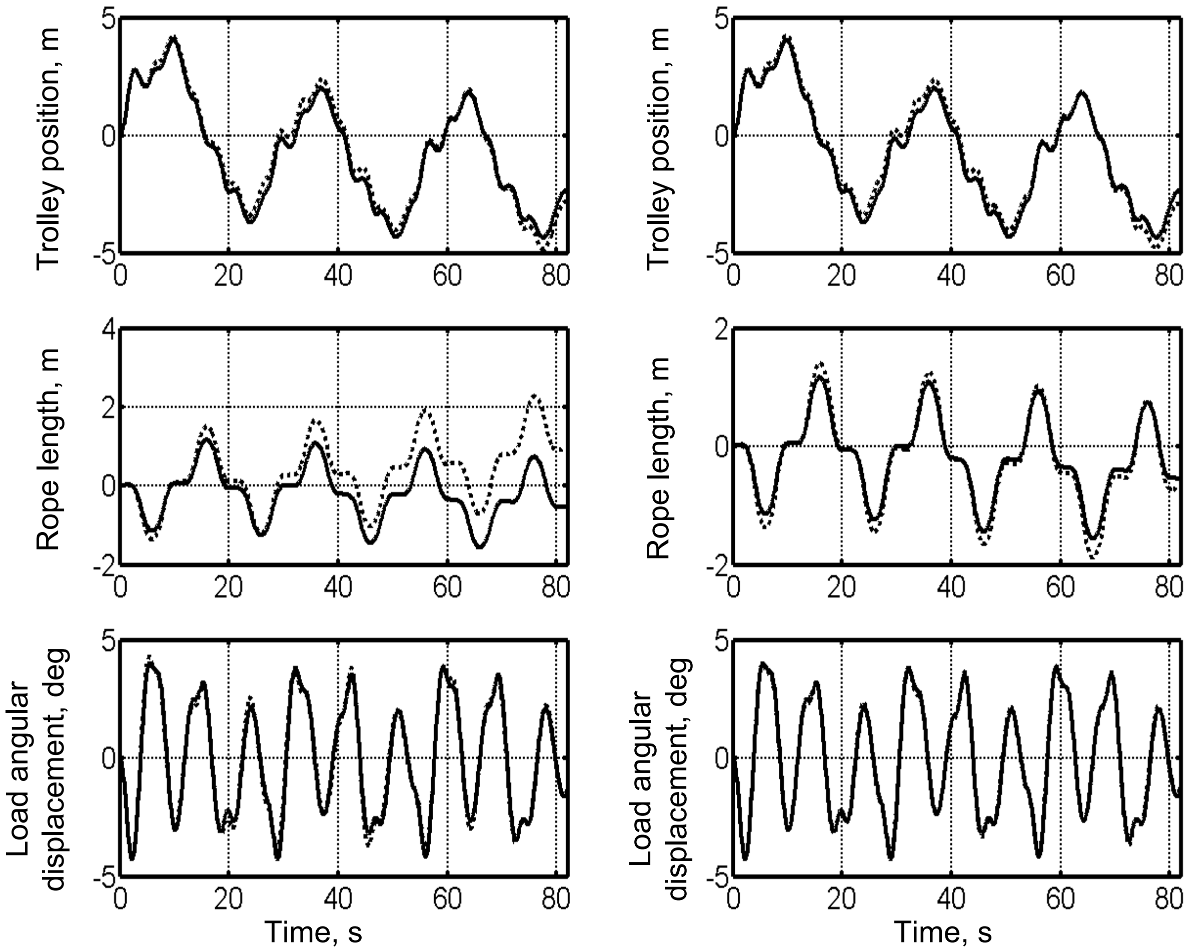

Figure 5.11. On the left: the results of the validation experiment (solid line – the identified model; dashed line – the data-based validation model). On the right: comparison of the identified model (solid line) and the analytically derived model (dotted line) with the inputs used in the validation experiment

In this situation it is difficult to obtain accurate estimates of the system matrices (see Section 4.2.1.1). Simulations show that increasing the sampling period does not help to move the poles away from the point $z= 1$ since the poles of the continuous plant are located on the imaginary axis or in a close proximity to it. The δ-operator approach and a longer prediction horizon (more than $3\ s$) may improve the identification results.

The situation is completely different if an asymptotically stable, though quite oscillatory, plant is identified, for example, the double-link pendulum [KT2002, KT2003] or the GC with the rope length being fixed by brakes (only two DOFs are retained). In that case it is possible to achieve an exceptionally high accuracy of the identified model, see Table 5.3 and Figure 5.12 (validation experiment).

Table 5.3. Poles of the identified and the analytically derived models of the GC with the fixed rope length

Poles of the analytically derived model	Poles of the identified model
0,987783 ± 0,09683885 j	0,987816 ± 0,0968010 j
1	0.999999
0,993305	0,993299

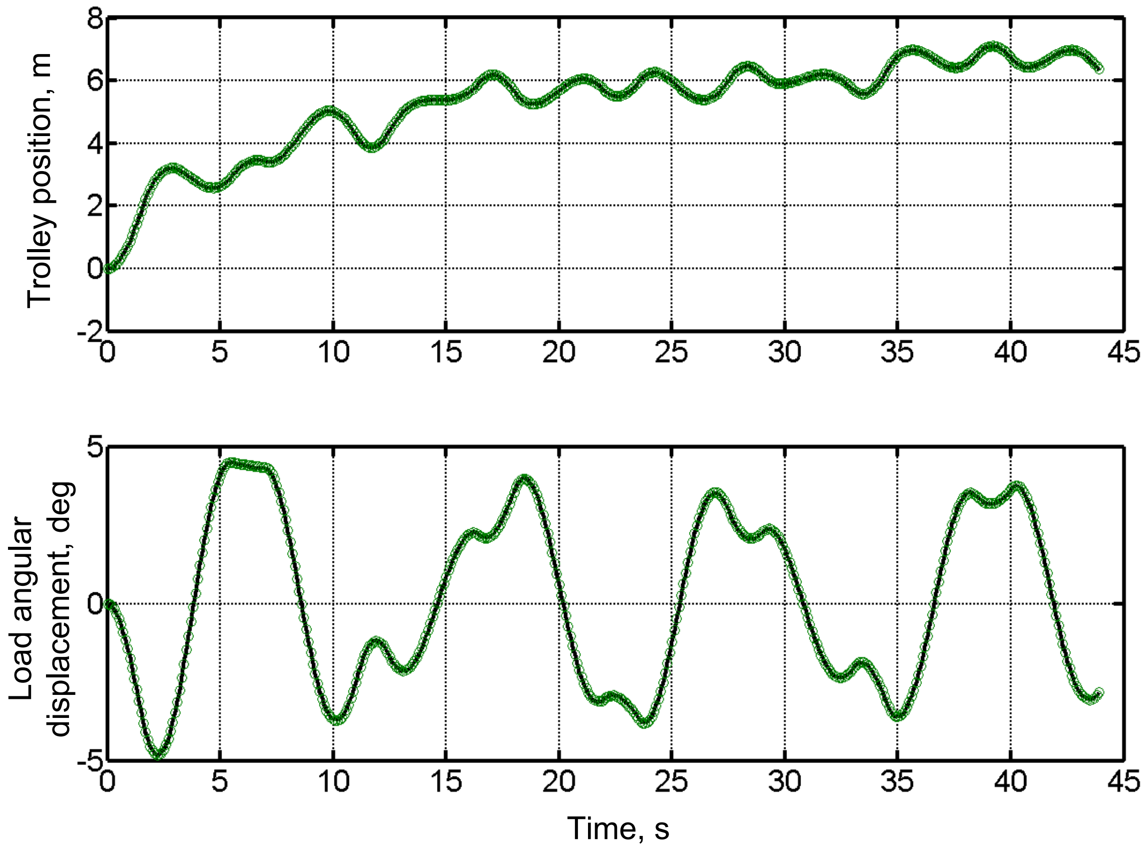

Figure 5.12. Results of the validation experiment for the GC with the fixed rope length (the rope length is 28 m). Unlike in the case with the free rope length, a very precise model can be identified (solid line); compare with Figure 5.11

In summary, the quality of identification depends on the stability of the plant. The best results can be achieved for asymptotically stable plants. The worst results are gained for unstable and marginally stable plants. This conclusion corresponds to the overall difficulties related to modeling and simulation of unstable systems [Chu2003].

5.6.4. Design of the control system

In this subsection we will design and analyze a controller for fine positioning. The GC is a MIMO system. And in the previous section we have identified a linear model of the GC with the completely measurable the state vector. Therefore, it makes sense to design a control system on the basis of the linear stationary full state feedback minimizing an LQ quality functional [LA1996, Bur2001]. To keep our example simple we ***intentionally*** consider (1) the simplest LQ control, (2) the case when the state vector is completely measurable[23] and (3) noise-free. However, this example is not impractical because the LQ control has quite significant robustness properties provided that the state vector is completely measurable and the controller is designed for the "true" model of the plant [SA1977]. Should the real control system with the noisy measurements and the incompletely measurable state vector be developed, the entire potential of the automatic control theory (including Kalman's observers [Zak2003]) can be employed in conjunction with the proposed DVP development method.

Consider the LQ quality functional J to be minimized:

$$J = \sum_{k=1}^{\infty} \left(x_k^{\,T} Q x_k + u_k^{\,T} R u_k \right) \to \min_{u} , \tag{5.19}$$

where x is the state vector, u is the control vector, Q and R are the design weight matrices. Let the matrices Q and R be diagonal[24]. Applying several trial-and-error iterations, we define the matrices as follows:

$$Q = diag\begin{pmatrix}1 & 1 & 10^4 & 1 & 1 & 10^2\end{pmatrix},$$
$$R = diag\begin{pmatrix}10^{-1} & 10^{-2}\end{pmatrix}.$$

This choice is explained by the relative scale of the phase variables θ_1, θ_2 and φ and their derivatives. The linear displacement of the trolley by 1 m corresponds to the angular rotation of the trolley drive by $\Delta\theta_1 = 1/1{,}44 \cdot 10^{-2} = 69{,}44\, rad$. The linear variation of the rope length by 1 m corresponds to $\Delta\theta_2 = 1/1{,}22 \cdot 10^{-2} = 81{,}97\, rad$. Yet, the angular displacement of the load by $\Delta\varphi = 0{,}1 rad$ corresponds to a substantial deflection angle of about 6°. So, the

---

[23] It well may be the case because all state variables have real physical meaning (angles and their derivatives)

[24] The LQ control (5.19) lacks the integral action and the system will have steady-state offsets for persistent disturbances (drifts, wind excitation, load mass variations). The integral component can be added by augmenting the state of the GC model [Mac1989]. Actually, the rope-load pendulum does not need the integral action (due to the asymptotically stability) and within the adopted GC reference model it is not possible to drive the load displacement angle to an arbitrary non-zero steady-state position using the trolley and hoist drives

variations of θ_1, θ_2 and φ may differ by 2-3 orders of magnitude. When defining matrix Q, we have to take into account the relative scale of the phase velocities and the fact that the angular displacement of the load is the primary controlled variable. The elements of the matrix R are chosen to scale down the influence of the control torques. The torque T_1 is of the order 10^2-10^3 and the torque T_2 is of the order 10^3-10^4, hence the elements of R.

Synthesizing the feedback matrix K with MATLAB / Control system toolbox function `dlqr` [MLAB2005], we obtain (six significant digits are retained)

$$K = \begin{pmatrix} 3.00977 & 0.241803 \\ -0.0527380 & 9.53668 \\ 1741.48 & -53.6705 \\ 7.93292 & 0.0809296 \\ 1.58774 & 28.7878 \\ 6212.08 & 165.761 \end{pmatrix}^T .$$

The control law has the following form: $u = K(r - x)$, where r is the state reference, i.e. the desired final value of the state vector.

The simulation of the designed control system is shown in Figures 5.13-5.14. The simulation program in IGRIP, which corresponds to Figure 5.14, is listed in Appendix 2 (`crane_control.gsl`).

Let us analyze the results. The fact that the simulation of the control system with the identified model is in line with the simulation with the nonlinear DVP implies that the quality of the identified model is fairly good[25]. But the robustness of the LQ controller may be insufficient despite the above-mentioned theoretical robustness properties. Thus, when the identified model is not accurate enough (the model is very sensitive because the poles of the plant are located close to the unit circle), the designed control law, which is being applied to the nonlinear DVP, has been found to be unstable. It means that the LQ method is indeed utterly sensitive to the inaccuracies of the model, see Section 4.2.1.2.

[25] The models are compared in a closed-loop and one can be more confident about the quality of the identified model than in case of the comparison in an open-loop, see Section 4.2.1.1

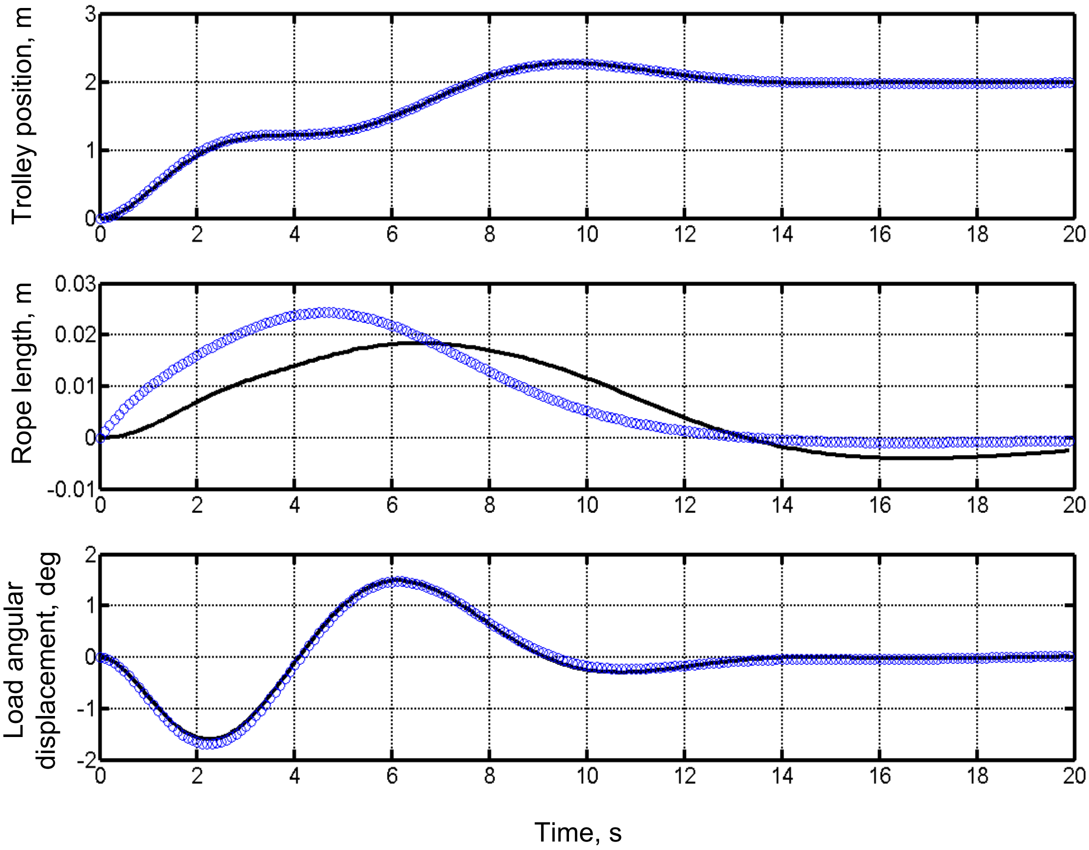

Figure 5.13. Simulation of the designed control system when the trolley is translated by 2 m. Solid line – simulation with the nonlinear DVP in IGRIP; circled line – simulation with the identified model in MATLAB. The rope length practically does not change. The largest discrepancy is observed for the rope length DOF, which has not been accurately identified (see also Figures 5.9 and 5.11)

As far as the control quality is concerned, the duration of the transient process of about 13 *s* is reasonable because the period of the natural oscillations of the load-rope assembly for the rope length of 28 *m* is $2\sqrt{28} \approx 11 s$. The only way to set the load in motion is to create a non-zero deflection angle between the rope and the vertical line. Since the rope is not rigid, it is difficult to affect the natural oscillations. So, if there is no overshoot of the load angular displacement at the end of the transient process, the duration of the process would be approximately equal to the period of the natural oscillations. This result is consistent with [SS1982] and the recommendations for the practical crane operations [Uns1976]. Here the load travel distance is not large and the duration of the transient process is about one full period of the natural oscillations.

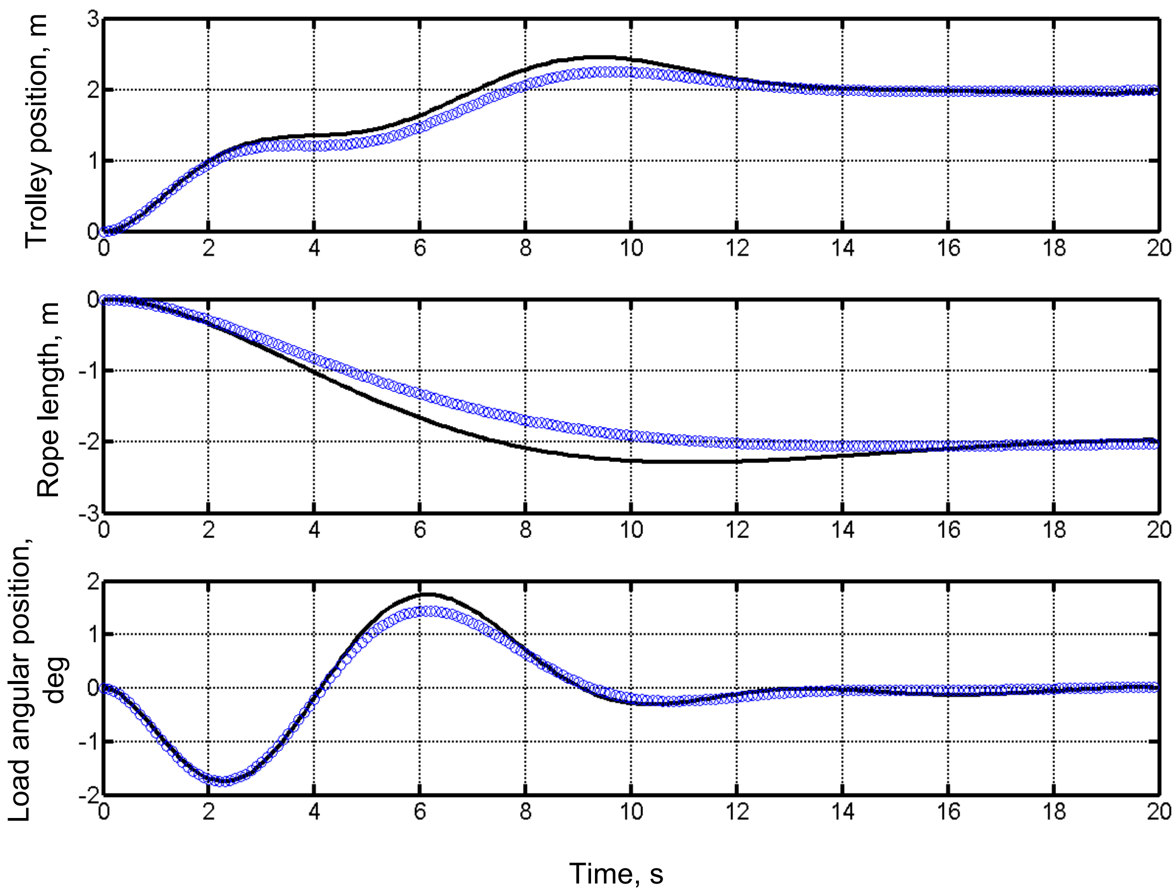

Figure 5.14. Simulation of the designed control system when the trolley is translated by 2 m and the load is hoisted by 2 m. Solid line – simulation with the nonlinear DVP in IGRIP; circled line – simulation with the identified model in MATLAB. A typical nonlinear effect can be observed: when the load is being hoisted, the amplitude of the load angular displacement is larger than when the rope length remains constant, compare with Figure 5.13. By contrast, when the load is being lowered, the amplitude becomes smaller

From the simulation data we can observe that the torques of the drive motors T_1 and T_2 have not been fully used. It means there is still a significant potential for reducing the duration of the transient process.

The controlled GC has a three times shorter fine positioning transient process than the GC without control (13 *s* vs. 45 *s*). Anyway, taking into account the mass of the moving elements (tens of tons), translating the trolley by 2-3 *m* in 10-12 *s* has to be considered as reasonably fast.

Theoretically, the duration of the transient process could be made much shorter if we (1) reformulate the control problem in such a way that the ***linear*** displacement of the load (relative to the destination point) would be the principal control variable and (2) allow large ***angular*** displacements of the load.

With the designed control system, the human operator's role is to assign the desired target points for fine positioning, to control the GC during large-range motions, and to keep an overall supervision of the GC operations. The automated fine positioning operations should facilitate the work of the GC operator.

The control system development process is usually finalized by performing exhaustive simulation experiments in the DVP system in a number of work regimes and with various sets of parameters (load mass, rope length) to assess the robustness (stability, performance) of the designed control system. We do not need to proceed further because our example has purely illustrative goals and the controller fine tuning belongs to the realm of the traditional methods of the automatic control theory and requires a direct reference to the real GC.

5.7. Summary

1. The development of the control system for fine positioning of a GC load has been considered to illustrate the proposed DVP control system development method. The GC is a complex plant due to its lightly damped structure, "inconvenient" place of exerting control, different types of stability for each DOF, and significant nonlinearities. It has been shown that the DVP systems appear to be rather appropriate for developing the GC control systems;
2. The control system development problem has been solved exclusively on the basis of the dynamic virtual prototyping involving neither the analytical modeling nor the GC physical embodiment. The controlled GC in a given work regime has a much shorter duration of the transient process that the uncontrolled GC;
3. The engineering applicability of the proposed development method has been proved. The results of the DVP-based method have been verified and validated by means of comparing them with the results obtained by employing the traditional methods.

CHAPTER 6. CONCLUSIONS

1. A new method of developing control systems on the basis of the DVP has been proposed and analyzed. The method facilitates the control system development process by means of (1) automating the transition from the conventional plant DVP to the dynamic model suitable for the control system design and (2) integrating the development process into the overall lifecycle. The proposed method requires neither the explicit analytical model nor the physical experiments with the real plant. Also, the method extends the applicability of the contemporary DVP/CALS systems to the development of control systems;

2. The DVP control system development method has been adapted to the selected class plants. The class includes the plants which are linearizable, quasi-stationary, stable or stabilizable without using the analytical model and have lumped parameters. The possibility of developing the desired control system with the DVP-based method for a plant, which belongs to the selected class, has been proved;

3. The specifics, limitations and rules of applying the existing mathematical methods of identification and control system design in the context of DVP have been systematically analyzed. The specifics arise from the fact that the DVP and the generated data-based model are employed instead of the analytical model or the real data. The following two major groups of methods have been considered: the methods, which involve the intermediate identification of the analytical model, and the data-based methods, which do not require such identification;

4. The ways of improving the quality and adequacy of the results have been discussed. The insufficiency of the information on the model uncertainty set and the lack of the formal MIMO control system design methods, which would not require the explicit knowledge of the uncertainty set structure, have been indicated as a bottleneck of the proposed DVP-based method. A procedure for constructing the general form of the uncertain plant on the basis of multiple simulations in the DVP system has been suggested;

5. The engineering applicability of the proposed development method has been proved by means of developing the control system for fine positioning of a gantry crane load;

6. The proposed DVP-based control system development method can be recommended to the users of the (D)VP/CALS systems, the control engineers, and the developers of the (D)VP/CALS systems. Moreover, the results obtained in this dissertation may be found useful in the education process for the contemporary CAD, CACSD, and (D)VP/CALS technologies.

CHAPTER 7. FUTURE DIRECTIONS

This dissertation concerns the method of applying DVP to the development of control systems and the adaptation of the method to the selected class of plants. However, some mathematical aspects of the method are deliberately set aside. Only when the method had been analyzed and elaborated have we understood the actual requirements to the mathematical methods for constructing DVPs, generating data-based models, and designing control systems. Hence, the directions of the future development of the method are foreseen in the following areas.

- Designing methods for the construction of DVPs that would allow to tackle the complexity of modeled systems. The VPs of real technical systems may contain a large number of parts and a massive amount of information. It is infeasible to construct the DVP and to model the dynamics unless the complexity of the system is somehow reduced. Specifically, the reduction method based on the kinematic graph of the system appears to be rather promising;
- Extending the DVP control system development to significantly nonlinear systems. For example:
    - Modeling and analysis (e.g. path following, fractal dimensionality) of chaotic systems with DVP rather than with analytical modeling seem to be a very fruitful direction of the future research. Such an analysis will help to identify and to avoid chaotic regimes and the regimes that are close to the bifurcation points;
    - Application of the hybrid automata theory will enable to model the nonstationary polyregime systems with abrupt changes of structure (e.g. the dynamics of a gantry crane as a whole with and without a container);
- Designing formal methods (especially state-space) for constructing the model uncertainty set for a class of parametric and nonparametric uncertainties on the basis of the DVP-generated data-based model. These methods will make it possible to address the robustness problems of the designed control systems. Specifically, the proposed procedure for constructing the general form of the uncertain plant on the basis of multiple DVP simulations has to be further elaborated, for instance, in terms of defining the grids and formulating the procedure in the time domain;
- Applying the DVP method to incorporate the properties (e.g. quantization, speed, delays, dynamics etc.) of real sensors, actuators and information processing hardware

and software in order to develop more realistic control systems. Also, applying DVP to functional diagnostics and integrating the DVP systems with SCADA (supervisory control and data acquisition) systems look quite attractive;

- Implementing the DVP control system development method as a commercial product, for example, in a form of plug-in to the contemporary DVP systems through the CATIA or STEP interface with the XML data-based model transfer to the CACSD systems.

As far as the gantry crane example is concerned, the model can be enhanced by taking into account the three-dimensional oscillations of the rope-spreader-load assembly, the additional joint between the rope and the spreader, the brakes and the real electro-mechanical characteristics of the drive motors. Such a detailed model has never been considered in the literature and, therefore, it would be useful for the crane engineers by itself. In addition, the control problem can be reformulated allowing larger variations of the rope angles in order to increase the dynamic performance of the fine positioning system.

The evolution of control system development resembles one of software development. In the early days of the computer era, programming was an extremely laborious and tricky process. The programmer had to have the knowledge of the computer hardware, the programming skills, and the understanding of the problem at hand. The emergence of compilers and high-level languages greatly simplified the process of programming as it became possible to concentrate on the problem and the program rather than the hardware. The recent concept of the model-driven architecture, which proclaims that "***a model generates the code***", has enabled to focus on the problem itself rather than the hardware or the programming (it its conventional sense).

By analogy, some fifty years ago the development of control systems used to be accomplished primarily with paper and a pen. The emergence of the computer brought the revolutionary tools of CACSD, which allow to concentrate on the control problem at hand rather than the implementation of the mathematical methods. In our opinion it is the DVP-based control system development that will play the role of the model-driven architecture in control engineering making the slogan "***a DVP generates the control system***" a reality.

BIBLIOGRAPHY

[AA2002] **Abedi, J., Aliabadi, S.,** *A new parallel finite element technique for simulation of floating objects* // IASTED conference on modelling and simulation, May 13-15, 2002, Marina del Rey, USA, pp. 264-270

[ABB2001] **Andrievsky, B.R., Blekhman, I.I., Bortsov, Yu.A., et al,** *Control of mechatronic vibrational systems (Upravleniye mekhatronnymi vibratsionnymi ustanovkami)*[1] / I.I. Blekhman I.I., A.L. Fradkov (eds.), St. Petersburg, Nauka, 2001, 278 p., (in Russian)

[Ade1996] *Adequate* / Webster's encyclopedic unabridged dictionary of the English language, New York, Gramercy books, 1996, p. 18

[AF2004] **Andrievskiy, B. R., Fradkov, A. L.,** *Control of chaos: Methods and applications. I. Methods* // Automation and remote control, 2003, Vol. 64, No. 5, pp. 673-713; Andrievskiy, B. R., Fradkov, A. L., *Control of chaos: Methods and applications. II. Applications* // Automation and remote control, 2004, Vol. 65, No. 4, pp. 505-533

[Aka1983] **Akaike, H.,** *Information measures and model selection* // Bulletin of the international statistical institute, 50, 1983, pp. 277-290

[AKN1998] **Afanas'ev, V.N., Kolmanovsky, V.B., Nosov, V.R.** *Mathematical theory of the control system design (Matematicheskaya teoriya konstruirovaniya sistem upravleniya),* 2nd ed., Moscow, Vysshaya shkola, 1998, 574 p., (in Russian)

[ANM2003] **Abdel-Rahman, E.M., Nayfeh, A.H., Masoud, Z.N.,** *Dynamics and control of cranes: A review* // Journal of vibration and control, Vol. 9, No. 7, July 2003, pp. 863-908 (Bibliography: 167 references)

[Ano1998] **Anokhin, P.K.,** *Cybernetics of functional systems: Selected works (Kibernetika funktsionalnykh sistem)*, Moscow, Medicine, 1998, 400 p., (in Russian). There is an English book: **Anokhin, P.K.,** *Biology and neurophysiology of the conditioned reflex and its role in adaptive behavior*, Elsevier, 1973, 592 p.

[1] The transliteration of the original Russian title is given in parentheses

[ANS2005] *ANSYS*, 2005 [www.ansys.com]

[Arn2004] **Arndt, C.,** *Information measures: Information and its description in science and engineering*, Springer, 2004, 547 p.

[AT1987] **Auernig, J.W., Troger, H.,** *Time optimal control of overhead cranes with hoisting of the load* // Automatica, Vol. 23, No. 4, July 1987, pp. 437-447

[Aut2005] *Autodesk: AutoCAD,* 2005, [www.autodesk.com]

[AW1995] **Åström, K., Wittenmark, B.,** *Adaptive control*, Addison-Wesley, 1995, 574 p.

[Bae2002] **Baer, T.,** *Product, process, resource and order integration (Daimler Chrysler)* // Delmia European user conference, Stuttgart, 2002, (PDF document)

[BCD2001] **Butts, K., Cook, J., Davey, C., *et al*,** *Automotive Powertrain controller development using CACSD* // Perspectives in control engineering / T. Samad (ed.), IEEE press, 2001, pp. 370-392

[BDS2003] **Bratukhin, A.G., Davydov, Yu.V., Surov, V.I., Bratukhin, V.A.,** *Lifecycle of the products, which are developed with CALS (Zhiznenniy tsikl izdeliy, sozdavaemykh na printsipakh CALS)* // Mobile technologies (Mobilnye tekhnologii), No. 1, 2003, p. 3-9, (in Russian)

[Bel2000] **Belousov, I.R.,** *Constructing the equations of dynamics for robots-manipulators (Formirovaniye uravneniy dinamiki robotov-manipulyatorov)* // Preprints of IPM No. 45, Institute for applied mathematics of M. Keldysh, Moscow, 2000, (Bibliography: 47 references), (cited November 2005), [http://library.keldysh.ru], (in Russian)

[Ber1968] **Bertalanffy von, L.,** *General system theory: Foundations, development, applications*, New York, 1973, 336 p.

[BG2000a] **Baker, C., Glad, S.,** *Maintaining a steady arm* // Nuclear engineering international, Vol. 45, No. 548, March 2000, pp. 30-31

[BG2000b] **Baker, C., Glad, S.,** *Simulation-driven design helps create vibration-free control system for a robot with a 30 foot long arm* // Strain, Vol. 36, No.

4, November 2000, pp. 165-167

[BHN2004] **Böling, J., Häggblom, K., Nyström, R.,** *Multivariable uncertainty estimation on multi-model output matching* // Journal of process control, Vol. 14, No. 3, 2004, pp. 293-304

[BJ2000] **Bauer, D., Jansson, M.,** *Analysis of the asymptotic properties of the MOESP type of subspace algorithms* // Automatica, Vol. 36, No. 4, 2000, pp. 497-509

[BKT2004] **Babko, L.V., Krassi, B.A., Tuominen, J.O.,** *Automation of the control systems development using virtual dynamic modeling (Avtomatizatsiya razrabotki sistem avtomaticheskogo upravleniya s ispol'zovaniyem virtual'nogo dinamicheskogo modelirovaniya)* // The 15th conference "Extreme robotics", Central research institute for robotics and technical cybernetics, St. Petersburg, Russia, April 6-7, 2004, pp. 342-347, (in Russian)

[BNF1987] **Butenin, N.V., Neymark, Yu.I., Fufayev, N.A.,** *Introduction to the theory of nonlinear oscillations (Vvedeniye v teoriyu nelineinykh kolebany)*, 2nd ed., Moscow, Nauka, 1987, 384 p., (in Russian)

[Bon2005] *Bond graphs,* 2005, [www.bondgraph.com]

[BPV1966] **Babuška, I., Prager, M., Vitasek, E.,** *Numerical processes in differential equations*, Prague SNTL – Publishers of technical literature, Prague, 1966, 351 p.

[Bri2001] **Britton, C.,** *IT architectures and middleware*, Addison-Wesley, 2001, 296 p.

[Bur2001] **Burns, R.S.,** *Advanced control engineering*, Butterworth-Heinemann, 2001, 450 p.

[Cat2000] *Catastrophe theory. Chaos. Fractals // Methods of classical and modern control theory (Metody klassicheskoi i sovremennoy teorii avtomaticheskogo upravleniya)*, Vol. 3, Moscow state technical university of N. Bauman, 2000, pp. 92-192, (in Russian)

[CDK2001] **Coulouris, G., Dollimore, J., Kindberg, T.,** *Distributed systems: concepts and design*, 3rd ed., Addison-Wesley, 2001, 772 p.

[Chu2003] **Chulichkov, A.I.,** *Mathematical models of nonlinear dynamics (Matematicheskie modeli nelineinoy dinamiki),* 2nd ed., Moscow, Fizmatlit, 2003, 296 p., (in Russian)

[Com1985] *Computer-aided control systems engineering* / M. Jamshidi and C.J. Herget (eds.), North-Holland, 1985, 384 p.

[COM2005] *COMSOL: Multiphysics / MEMS*, 2005, [www.comsol.com]

[Cou1956] **Couffignal, L.,** *Essay d'une définition générale de la cybernétique* // The 1st international congress on cybernetics, Namur (Belgium), June 26-29, 1956, Gauthier-Villars, Paris, 1958, pp. 46-54, (in French)

[CP2004a] **Chiusso, A., Picci, G.,** *On the ill-conditioning of subspace identification with inputs* // Automatica, Vol. 40, No. 4, 2004, pp. 575-589

[CP2004b] **Chiusso, A., Picci, G.,** *Numerical conditioning and asymptotic variance of subspace estimates* // Automatica, Vol. 40, No. 4, 2004, pp. 677-683

[Cyb1994] *Cybernetics* // McGraw-Hill dictionary of scientific and technical terms, 5th ed., S.P. Parker (ed.), McGraw-Hill, 1994, p. 504

[Das2005] *Dassault Systèmes: CATIA, V5,* 2005, [www.3ds.com]

[Dei1979] **Deiych, A.M.,** *Identification of dynamic systems (Metody identifikatsii dinamicheskikh ob'yektov),* Moscow, Energiya, 1979, 240 p. (in Russian)

[Del2005] *Delmia: IGRIP, ENVISION, V5,* 2005, [www.delmia.com]. Also: IGRIP user manual (electronic document, not freely available)

[DFT1990] **Doyle, J., Francis, B., Tannenbaum, A.,** *Feedback control theory*, Macmillan publishing co., 1990, 202 p.

[Dif2005] *Differential equations with symbolic computation* / D. Wang, Zh. Zheng (eds.), Springer, 2005, 374 p.

[DK2001] **Diduk, N.N., Koval, V.N.,** *Does the science of cybernetics exist? (about the role of cybernetics in natural science)* // Journal of automation and information science, Vol. 33, No. 6, 2001, Publisher: [www.begellhouse.com]

[DS1979] **Doyle, J., Stein, G.,** *Robustness with observers* // IEEE transactions on automatic control, Vol. AC-24, No. 4, 1979, pp. 607-611

[Dym2005] *Dymola*, 2005, [www.dynasim.se]

[Dyn1978] *Dynamic modeling and testing of technical systems (Dinamicheskoe modelirovanie i ispytaniya tekhnicheskikh sistem)* // Kochubievsky I.D., Strazhmeister V.A., Kalinovskaya L.V., Matveev P.A., Moscow, Energia, 1978, 303 p., (in Russian)

[Eff1989] *Efficiency of complex systems. Dynamic models (Effektivnost' slozhnykh sistem. Dinamicheskie modeli)* / Vinogradov V.A., Gruschansky V.A., Dovgush S.I. *et al*, Moscow, Nauka, 1989, 285 p., (in Russian)

[EHDS2005] **Erleben, K., Henriksen, K., Dohlmann, K., Sporring, J.,** *Physics-based animation*, Delmar Thomson learning, 2005, 800 p.

[EK2003] **Ellermann, K., Kreuzer, E.,** *Nonlinear dynamics in the motion of floating cranes* // Multibody system dynamics, No. 9, 2003, pp. 377-387

[EKM2002] **Ellermann, K., Kreuzer, E., Markiewicz, M.,** *Nonlinear dynamics of floating cranes* // Nonlinear dynamics, No. 27, 2002, pp. 107-183

[EME1992] **Ebeid, A.M., Moustafa, K.A., Emara-Shabaik, H.E.,** *Electromechanical modeling of overhead cranes* // International journal of system science, Vol. 23, No. 2, 1992, pp. 2155-2169

[Eng1883] **Engels, F.,** *Dialectics of nature,* 1883, (cited: November 2005), [www.marxists.org]

[FDGV1998a] **Favoreel, W., De Moor, B., Van Overschee, P., Gevers, M.,** *Model-free subspace-based LQG design* // American control conference, 1998, Vol. 5, pp. 3372 –3376

[FDGV1998b] **Favoreel, W., De Moor, B., Gevers, M., Van Overschee, P.,** *Closed loop model-free subspace-based LQG design* // IEEE Mediterranean conference on control and automation, Haifa, Israel, 1998, TP5-4

[Fed2004] **Fedosov, B.T.,** *Studying and modeling dynamic virtual objects (Issledovanie nekotorykh dinamicheskikh kharakteristik virtualnogo sportivnogo avtomobilya v komp'yuternoy igre Need For Speed 3 i ego modelirovanie)*, (cited: November 2005), [http://model.exponenta.ru/bt/bt_0009.html], (in Russian)

[FEM2002] *FEMAS: Developing future electronics microassembly systems*, Helsinki

University of Technology, 2002, (cited: November 2005), [www.control.hut.fi/Research/femas]

[Fre1935] **Freundlich, H.,** *Thixotropy*, Paris, Hermann, 1935, 50 p.

[Fun2001] *Fundamentals of the control design theory for the high-precision missile systems of land forces* / B.G. Gursky, M.A. Luschanov, E.P. Spirin *et al*, V.L. Solunin (ed.), Moscow state technical university of N. Bauman, Moscow, 2001, 328 p., (in Russian)

[Fuz1999] *Fuzzy logic control: Advances in applications* / R. Babuška (ed.), World scientific series in robotics and intelligent systems, Vol. 23, World scientific, 1999, 329 p.

[Gai2001] **Gaikovich, A.I.,** *The fundamentals of the theory for complex technical system design (Osnovy teorii proektirovaniya slozhnykh tekhnicheskikh sistem)*, St. Ptersburg, Morintekh, 2001, 432 p., (in Russian)

[Gen2004] *General Motors buys IBM supercomputer for design, crash test work* / The Associated Press, April 22, 2004 // USA Today, (cited: November 2005), [www.usatoday.com]

[GK2004] **Gavrilov, S.V., Konoplev V.A.,** *Computer methods for analyzing multi-link mechatronic systems (Kompyuternye tekhnologii issledovaniya mnogozvennykh mekhatronnykh sistem)*, St. Petersburg, Nauka, 2004, 191 p., (in Russian)

[Gru2001] **Grübel, G.,** *Computer-automated control system design* // Perspectives in control engineering / T. Samad (ed.), IEEE Press, 2001, pp. 42-70

[Gus2000] **Gustafsson, F.,** *Adaptive filtering and change detection*, John Wiley & Sons, 2000, 500 p.

[GV1989] **Golub, G.H., Van Loan, C.F.,** *Matrix computations*, 2nd ed., Baltimore and London, The John Hopkins University Press, 1989, 642 p.

[HA1996] **Hägglund, T., Åström, K.,** *Automatic tuning of PID controllers* // The control handbook / W. Levine (ed.), CRC Press, FL, 1996, pp. 817-826

[HC1999] **Huang-Haodong, Chen-Zongji,** *Virtual prototyping and virtual prototype in control system* // Journal of Beijing university of aeronautics and astronautics, June 1999, Vol. 25, No. 3, pp. 284-287, (in Chinese, reference and English abstract by INSPEC)

[HK2000] **Huang, D., Katayama, T.,** *A subspace identification of continuous-time systems using delta-operator model* // Proceedings of the 31st international symposium on stochastic systems theory and its applications (Japan), 2000, pp. 13-18, (reference and abstract by INSPEC)

[HK2001] **Huang, D., Katayama, T.,** *A subspace-based method for continuous-time model identification by using delta-operator model* // Transactions of the institute of systems, control and information engineers (Japan), January 2001, Vol. 14, No. 1, pp. 1-9, (reference and abstract by INSPEC)

[HMB1993] **Harris, C.J., Moore, C.G., Brown, M.,** *Intelligent control: Aspects of fuzzy logic and neural nets*, World series in robotics and automated systems, Vol. 6, 1993, 400 p.

[HSB2002] **Hindi, H., Seong, C.-Y., Boyd, S.,** *Computing optimal models from frequency domain data* // IEEE 41st conference on decision and control, USA, December 2002, pp. 2898-2905

[Hyo2001] **Hyötyniemi, H.,** *Multivariate regression: Techniques and tools*, Helsinki University of Technology, Control engineering laboratory report 125, 2001, 207 p., [www.control.hut.fi/rpt/r125isbn9512255871.pdf]

[Hyo2003] **Hyötyniemi, H.,** *Emergence and complex systems: Towards new practices for industrial automation* // Intelligent processing and manufacturing of materials, Sendai, Japan, May 18 - 23, 2003

[IBM2005] *IBM-CATIA. Product synthesis: analysis*, [www.ibm.com/catia]

[Inf2003] *Information technology for the product lifecycle management.* Proceedings of the international conference, November 25-26, 2003, Central research institute of robotics and technical cybernetics, St. Petersburg, 135 p., (in Russian), [www.rtc.ru], (cited: November 2005)

[Ise2005] **Isermann, R.,** *Mechatronic systems fundamentals,* Springer, 2005, 624 p.

[ISO1981] *ISO 4302-1981. Cranes – wind load assessment* / International organization for standardization, 1st ed., 1981

[ISO1995] *ISO 668-1995. Series 1 freight containers. Classification, dimensions and ratings* / International organization for standardization, 1995

[ISO2005] *ISO 10303-*. STEP – Standard for exchange of product data* / International organization for standardization, 2005, [www.iso.org]

[Ite2002] *Iterative identification and control* / Albertos, P., Sala, A. (eds.), Springer Verlag, 2002, 309 p.

[JK1980] **Jakoby, S., Kowalik, J.,** *Mathematical modeling with computers*, Prentice-Hall, 1980, 292 p.

[KB2003] **Krassi, B.A., Babko, L.V.,** *Using virtual modeling systems in the process of development of automatic control systems (Ispol'zovaniye sistem virtualnogo modelirovaniya v protsesse razrabotki SAU)* // The 32nd week of science: Proceedings of inter-university scientific conference, Vol. 5, St. Petersburg state technical university, St. Petersburg, Russia, November 24-29, 2003, pp. 53-54, (in Russian)

[KBTD2000] **Karkoub, M., Balas, G., Tamma, K., Donath, M.,** *Robust control of flexible manipulators via μ-synthesis* // Control engineering practice, Vol. 8, No. 7, 2000, pp. 725-734

[KCI2001] *KCI Konecranes International Plc: Technical specification of a 66 ton gantry crane,* 2001 [2]

[KKP2005] **Katayama,T., Kawauchi, H., Picci, G.,** *Subspace identification of closed loop systems by the orthogonal decomposition method* // Automatica, Vol. 41, No. 5, 2005, pp. 863-872

[KL2001] **Keviczky, T., Lantos, B.,** *Subspace-based multivariable identification and robust control of the MIT SERC interferometer testbed* // IEEE instrumentation and measurement conference, Hungary, May 2001, pp. 20-25

[KM2001] **Kim, C., Mauborgne, R.,** *Creating new market space* // Harvard business review on innovation, Harvard business school press, 2001, pp. 1-30

[KMN1989] **Kahaner, D., Moler, C., Nash, S.,** *Numerical methods and software*, Prentice-Hall, 1989, 495 p.

[KMR1990] **Karnopp, D., Margolis, D., Rosenberg, R.,** *System dynamics: A unified*

[2] By the courtesy of Mr. Hannu Oja, KCI (2004)

approach, New York, Wiley InterScience, 1990

[KO1994] **Keen, S.D., Olson, W.W.,** *Anti-lock brake simulation and vehicle modeling using DADS* // International mechanical engineering congress and exposition, November 1994, pp. 293-300

[Koi2000] **Koivo, H.,** *Soft computing in dynamical systems.* Lecture notes, Helsinki University of Technology, 2000, 123 p., [www.control.hut.fi/Kurssit/AS-74.3115/Material] (cited: November 2005, PDF document)

[Kol1994] **Kolesnikov, A.A.,** *Synergetic control theory: Invariants, optimization, synthesis (Sinergeticheskaya teoriya upravleniya: Invarianty, optimizatsiya, sintez)*, Taganrog-Moscow, 1994, 344 p., (in Russian)

[Kro1999] **Krouse, J.,** *Smarter to market* // Adams automotive solutions, 1999, [www.adams.com/solutions/auto], (cited: November 2005)

[KT2002] **Krassi, B.A., Tuominen, J.O.,** *Virtual prototyping for automated generation of mathematical models* // IASTED international conference on modelling and simulation, Marina del Rey, CA, USA, May 13-15, 2002, pp. 287-291

[KT2003] **Krassi, B.A., Tuominen, J.O.,** *Virtual prototyping for integration of the control system design chain* // IASTED international conference on modelling and simulation, Palm Springs, CA, USA, February 24-26, 2003, pp. 99-105

[KZ2001] **Karkoub, M.A., Zribi, M.,** *Robust control schemes for an overhead crane* // Journal of vibration and control, No. 7, 2001, pp. 395-416

[LA1996] **Lublin, L., Athans, M.,** *Linear quadratic regulator control* // The control handbook / W. Levine (ed.), CRC Press, FL, 1996, pp. 635-650

[Lan2001] **Landau, I.D.,** *Identification in the closed loop: A powerful tool (better design models, simpler controllers)* // Control engineering practice, No. 9, 2001, pp. 51-65

[LBA1998a] **Losino-Losinskiy, G.E., Bratukhin, A.G., Arutyunov, S.G., Dmitrov, V.I.,** *System technology of the virtual enterprise and the experience of developing "Buran" (Sistemnaya tekhnologiya virtualnogo predpriyatiya na osnove opyta organizatsii rabot po sozdaniyu "Burana")* // Airspace systems and technology (Aviakosmicheskaya tekhnika i tekhnologiya), No. 4, 1998, pp. 3-5, (in Russian)

[LBA1998b] **Losino-Losinskiy, G.E., Bratukhin, A.G., Arutyunov, S.G., Dmitrov, V.I.,** *Conceptual design of the virtual enterprises with the experience of developing the spacecraft "Buran" (Kontseptualnoe proektirovaniye virtualnykh predpriyatiy na osnove ispolzovaniya opyta sozdaniya kosmicheskogo korablya "Buran")* // Bulletin of machine engineering (Vestnik mashinostroeniya), No. 11, 1998, pp. 3-10, (in Russian)

[LBA1999] **Losino-Losinskiy, G.E., Bratukhin, A.G., Arutyunov, S.G., Dmitrov, V.I.,** *Scientific bases and implementation of the virtual enterprise in the process of developing the spacecraft "Buran" (Sozdaniye nauchnyh osnov i vnedreniya virtualnogo predpriyatiya pri realizatsii proekta kosmicheskogo korablya "Buran")* // Flight (Polyet), No. 8, 1999, pp. 26-30, (in Russian)

[LEG2005] *LEGO*, 2005, [www.lego.com]

[LGA1996] **Lublin, L., Grocott, S., Athans, M.,** *H_2 (LQG) and H_∞ control* // The control handbook / W. Levine (ed.), CRC Press, FL, 1996, pp. 651-661

[LGV1998] **Lovera, M., Gustafsson, T., Verhaegen, M.,** Recursive subspace identification of linear and non-linear Wiener state space models // Automatica, Vol. 36, No. 11, 2000, pp. 1639-1650

[Lju1996] **Ljung, L.,** *System identification* // The control handbook / W. Levine (ed.), CRC Press, FL, 1996, pp. 1033-1054

[Lju1999] **Ljung, L.,** *System identification: Theory for the user*, 2nd ed., Prentice Hall, 1999, 384 p.

[LL1996] **Langer, J., Landau, I.D.,** *Improvement of robust digital control by identification in the closed loop. Application to a 360° flexible arm* // Control engineering practice, Vol. 4, No. 12, 1996, pp. 1637-1646

[LM1996] **Ljung, L., McKelvey, T.,** Subspace identification from clod loop data // Signal processing, Vol. 52, No. 2, 1996, pp. 209-215

[LMS2001] *LMS International: Virtual.Lab,* (cited: November 2005), [www.lms.be]. Articles: Baker, C., Glad, S. LMS DADS creates vibration free robot arm

(2001) [Cases > Virtual Prototyping Cases > Dynamic Motion Simulation], (HTML document). Design reliable control systems validated on realistic mechanism models (2003) [LMS Virtual.Lab Motion], (PDF document)

[LMS2005] *LMS News*, May 2005, 40 p.

[LS1995] **Lewis, F.L., Syrmos, W.L.,** *Optimal control*, 2nd ed., John Wiley & Sons, 1995, 541 p.

[MA1998] **Mäntylä, M., Andersin, H.,** *Enterprise systems integration*, Helsinki, HUT, 1998, 313 p.

[Mac1989] **Maciejowski, J.M.,** *Multivariable feedback design*, Addison-Wesley, 1989, 424 p.

[Map2005] *Maplesoft: MAPLE*, 2005, [www.maplesoft.com]

[Mas2003] **Mason, H.,** *ISO 10303: A key standard for the global market* // ISO Bulletin, January 2002, (cited: November 2005), [www.iso.org]

[MATH2005] *Mathematica*, 2005, [www.wolfram.com]

[MAX2005] *3D MAX,* 2005, [www.3dmax.com]

[MCA2005] *MODELICA,* 2005, [www.modelica.org]

[MCAD2005] *Mathsoft: Mathcad,* 2005, [www.mathsoft.com]

[MEO1998] **Mattison, S.E., Elmqvist, H., Otter, M.,** *Physical system modeling with Modelica* // Control engineering practice, No. 6, 1998, pp. 501-510

[MG1990] **Middleton, R.H., Goodwin, G.C.,** *Digital control and estimation: a unified approach*, Prentice Hall, 1990, 538 p.

[ML2005] **Michaels, D., Lunsford, J.L.,** *Boeing, Airbus look to car companies. Methods to speed up jetliner production* // The Wall Street journal, April 1, 2005, p. B1

[MLAB2005] *The MathWorks: MATLAB, Control, Identification, Symbolic Mathematics, Signal Processing toolboxes, Simulink and SimMechanics,* 2005, [www.mathworks.com]

[MM2005] *MathModelica*, 2005, [www.mathcore.com]

[MN2003] **Masoud, Z.N., Nayfeh, A.H.,** *Sway reduction on container cranes using delayed feedback controller* // Nonlinear dynamics, No. 34, 2003, pp. 347-358

[Mod1974] *Modeling* / B.V. Biryukov, Yu.A. Gastev, E.S. Geller // The large soviet encyclopedia, 3rd ed., Vol. 16, Moscow, Soviet encyclopedia, 1974, pp. 393-395

[Mod1994] *Model* // McGraw-Hill dictionary of scientific and technical terms, 5th ed., McGraw-Hill, 1994, p. 1283

[MRX2005] *National Instruments: MATRIXx,* 2005, [www.ni.com]

[MSC2005] *MSC Software: Virtual product development tools, SimOffice (MSC.ADAMS, MSC.Nastran etc.),* 2005, [www.mscsoftware.com]

[MT1975] **Mesarovic, M.D., Takahara, Y.,** *General systems theory: Mathematical foundations*, Academic press, 1975, 268 p.

[Nad1990] **Nadel, L.,** *Sixth sense: The whole-brain book of intuition, hunchies, gut feelings, and their place in your everyday life*, Prentice Hall, 1990, 234 p.

[NY2001] **Novoseltsev, V.N., Yashin, A.I.,** *New horizons in mathematical modeling (Novye gorizonty matematicheskogo modelirovaniya)* // International conference on “Parallel computing and control”, Institute of control problems of Russian academy of sciences, 2001, pp. 214-244

[OCT2005] *Octave*, 2005, [www.octave.org]

[OHH1999] **Oguamanam, D.C.D., Hansen, J.S., Heppler, G.R.,** *Dynamic response of an overhead crane system* // Journal of sound and vibration, Vol. 213, No. 5, 1999, pp. 889-906

[Ost1984] **Ostermeyer, G.P.,** *Controlled dynamic systems modeling* // NATO ASI series, Series F: Computer & systems sciences, Vol. 9, 1984, pp. 415-418

[Pat2002] *Patent WO0222488*, KCI Konecranes International Plc / Oja, H., Rossi, H., Soederena, S., Toivonen, J., Arrangement for placing auxiliary ropes of crane hoist gears, Finland, March 21, 2002, (cited: November 2005), [http://fi.espacenet.com]

[Phe1996] *Phenomenology* // Webster’s encyclopedic unabridged dictionary of the

English language, New York, Gramercy books, 1996, p. 1080

[PID2001] *PID control*. Tutorial // Control engineering practice, Vol. 9, No. 11, 2000, pp. 1159-1263

[Pin2002] **Pintelon, R.,** *Frequency-domain subspace system identification using non-parametric noise models* // Automatica, Vol. 38, No. 8, 2002, pp. 1295-1311

[PKE1974] **Pugachev V.S., Kazakov, I.E., Evlanov, L.G.,** *Fundamentals of the statistical theory of the automatic systems* (*Osnovy statisticheskoy teorii avtomaticheskikh sistem*), Moscow, Mashinostroenie, 1974, 400 p., (in Russian)

[PLL2001] **Putov, V.V., Lebedev, G.N., Lebedev, V.V., Kozlov, Yu.K.,** *Adaptive-neural approach to controlling load oscillations transported at the external suspension of a helicopter (Adaptivno-neironniy podkhod v zadachakh upravleniya kolebaniyami gruzov, transportiruyemykh na vneshney podveske vertoleta)* // Proceedings of the 12th conference "Extreme Robotics", Central research institute for robotics and technical cybernetics, St. Petersburg, Russia, 2004, pp. 267-274, (in Russian)

[Pod2002] **Podchufarov, Yu.B.,** *Physical and mathematical modeling of control systems (Fiziko-matematicheskoye modelirovaniye system upravleniya i kompleksov)* / A.G. Shipunov (Ed.), Moscow, Fizmatlit, 2002, 168 p., (in Russian)

[PRO2005] *ProENGINEER,* 2005 [www.ptc.com]

[PS1985] **Prigogine, I.R., Stengers, I.,** *Man's new dialogue with nature*, London, 1985, 349 p.

[PT1996] **Phillips, G.M., Taylor P.J.,** *Theory and applications of numerical analysis*, 2nd ed., Academic press, 1996, 447 p.

[PTV1999] **Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.,** *Numerical recipes in C: The art of scientific computing*, 2nd ed., Cambridge university press, 1999, 994 p.

[Pug1979] **Pugachev V.S.,** *Probability theory and mathematical statistics for engineers*, Oxford Pergamon, 1984. The original version has been used:

Pugachev, V.S., *Teoriya veroyatnostey i matematicheskaya statistika*, Moscow, Nauka, 1979, 496 p., (in Russian)

[PY2001] **Polonnikov, R.I., Yusupov, R.M.** *Will the 21st century accept cybernetics?* // Journal of automation and information science, Vol. 33, No. 12, 2001, Publisher: [www.begellhouse.com]

[RGL2002] **Reinelt, W., Garulli, A., Ljung, L.,** *Comparing different approaches to model error modeling in robust identification* // Automatica, Vol. 38., No. 5, 2002, pp. 787-803

[Ris1978] **Rissanen, J.,** *Modelling by shortest data description* // Automatica, Vol. 14, No. 5, 1978, pp. 465-471

[Ris2001] **Rissanen, J.,** *Lectures on statistical modeling theory*, 2001, 65 p., [www.mdl-research.org], (cited: November 2005, PDF document)

[RJ1996] **Rimvall, C. M., Jobling, C. P.,** *Computer-aided control system design* // The control handbook / W. Levine (Ed.), CRC Press, 1996, pp. 429-442

[RJ2004] **Ren-Weiqun, Jin-Guodong,** *Active suspension design for reducing the vehicle-generated road damage* // China mechanical engineering, 15(22), 2004, pp. 2059-2063, (reference and abstract by INSPEC)

[Rob2002a] **Roberts, E.,** *Virtual assembly & integration preparation in space systems (Astrium)* // Delmia European user conference, Stuttgart, 2002, (PDF document)

[Rob2002b] *Robust control: The parameter space approach* / J. Ackermann et. al., 2nd ed., Springer, 2002, 483 p

[Rob2003] **Robinson, B.,** *Getting bigger and faster: Ship to shore cranes* // Cargo systems, July-August 2003, pp. 31-34, (cited: November 2005), [www.csxworldterminals.com]

[SA1977] **Safonov, M., Athans, M.,** *Gain and phase margin for multiloop regulators* // IEEE transactions on automatic control, Vol. AC-22, No. 2, 1977, pp. 173-179

[Sam2001] **Samad, T.,** *Control systems for complexity management* // Perspectives in control engineering / T. Samad (ed.), IEEE press, 2001, pp. 218-238

[SATTS1993] **Shirai, Y., Arakawa, H.; Toda, N., Taneda, Y., Sakura, K.,** *Active vibration control for aircraft wing* // JSME international journal, Series C: Dynamics, control, robotics, design and manufacturing, Vol. 36, No. 3, 1993, pp. 319-324

[Sen2004] **Senichenkov, Yu.B.,** *Numerical simulation of hybrid systems,* St. Petersburg state polytechnical university, St. Petersburg, 2004, 206 p., (in Russian)

[SFS1977a] *SFS 4020. Rules for the design of hoisting appliances. Steel structures* / Finnish standards association SFS, 1977, (in Finnish)

[SFS1977b] *SFS 4020. Rules for the design of hoisting appliances. Calculation of loads due to acceleration of horizontal motions* / Finnish standards association SFS, 1977, (in Finnish)

[SFS2004] *SFS-EN 13135-1. Cranes. Safety. Design. Requirements for equipment. Part 1: Electrotechnical equipment* / Finnish standards association SFS, 2004, (in English)

[Sha1996] **Shamma, J.S.,** *Linearization and gain-scheduling* // The control handbook / W. Levine (ed.), CRC Press, FL, 1996, pp. 388-396

[SLAB2005] *Scilab,* 2005, [www.scilab.org]

[SM1990] **Sayers, M.W.; Mousseau, C.W.,** *Real-time vehicle dynamic simulation obtained with a symbolic multibody program* // Winter annual meeting of the American society of mechanical engineers, Nov. 1990, pp. 51-58

[SM2001] **Samarskii A.A., Mikhaiov, A.P.,** *Principles of mathematical modelling: Ideas, methods, examples,* Taylor & Francis, 2001, 360 p.

[Sod2002] **Söderström, T.,** *Model validation and model structure determination* // Circuits, systems, signal processing, Vol. 21, No. 1, 2002, pp. 83-90

[SP1996] **Skogestad, S., Postlethwaite, I.,** *Multivariable feedback control: Analysis and design*, John Wiley & Sons, 1996, 559 p.

[SS1982] **Sakawa, Y., Shindo, Y.,** *Optimal control of container cranes* // Automatica, Vol. 18, No. 3, 1982, pp. 257-266

[SSH1996] **Santina, M., Stubberud, A., Hostetter, G.,** *Sample-rate selection* // The control handbook / W. Levine (ed.), CRC Press, FL, 1996, pp. 313-321

[Ste2000] **Stepanov, M.F.** *Automatic solution of the formalized control problems (Avtomaticheskoe resheniye formalizovannykh zadach teorii avtomaticheskogo upravleniya)*, Saratov state technical university, 2000, 376 p., (in Russian)

[Ste2002] **Stenerson, J.,** *Industrial automation and process control,* Prentice Hall, 2002, 432 p.

[TCA1971] **Takahashi, Y., Chan, C.S., Auslander, D.M.,** *Parametereinstellung bei linearen DDC-Algorithmen* // Regelungstechnik und Prozeß Datenverarbeitung, No. 6, 1971, pp. 237-244, (in German)

[Tik1995] **Tikhonov, A.N.,** *Numerical methods for the solution of ill-posed problems*, Kluwer academic publishers, 1995, 268 p.

[TK1998] **Tanaka, S., Koundo, S.,** *Automatic measurement and control of the attitude of crane lifters. Lifter-attitude measurement and control* // Control engineering practice, No. 6, 1998, pp. 1099-1107

[Toi2000] **Toivonen, H.,** *Robust control methods* (lecture notes), Åbo Academi University, 2000, 88 p., [www.abo.fi/~htoivone/advcont.html] (cited: November 2005, printed and PDF document)

[TS1999] **Teryaev, E.D., Shamrikov, B.M.,** *Digital systems and stepwise adaptive control (Tsifrovye sistemy i poetapnoe adaptivnoe upravleniye)*, Moscow, Nauka, 1999, 330 p., (in Russian)

[Tsa1999] **Tsai, L.-W.,** *Robot analysis: The mechanics of serial and parallel manipulators*, John Wiley & Sons, 1999, 505 p.

[UBMS2000] **Uhl, T., Bojko, T., Mrozek, Z., Szwabowski, W.,** *Rapid prototyping of mechatronic systems* // Journal of theoretical and applied mechanics (Poland), Vol. 38, No. 3, 2000, pp. 645-668

[UGS2005] *Unigraphics NX*, 2005, [www.ugs.com]

[Uns1976] *Unstetigförderer 1* / Pajer, G., Scheffler, M., Keilhorn, H., Adam G., Kurth, F., VEB Verlag Technik, Berlin, 1976; *Unstetigförderer 2* / Scheffler, M., Dresig, H., Kurth, F., VEB Verlag Technik, Berlin, 1977 (in German). The Russian translation of 1981 has been used

[Usp1982] **Uspensky, V.A.,** *Gödel's incompleteness theorem,* Moscow, Nauka, 1982, 112 p., (in Russian)

[VD1996] **Van Overschee, P., De Moor, B.,** *Subspace identification for linear systems*, Kluwer academic publishers, 1996, 254 p.

[Ver1993] **Verhaegen, M.,** *Application of a subspace model identification technique to identify LTI systems operating in closed-loop* // Automatica, Vol. 29, No. 4, 1993, pp. 1027-1040

[Ver2002] **Verschoof, J.,** *Cranes – design, practice, and maintenance*, 2nd ed., Professional engineering publishing, London & Bury St. Edmunds, UK, 2002, 328 p.

[VHJ1996] **Varsamidis, T., Hope, S., Jobling, C.,** *An object-oriented information model for computer-aided control engineering* // Control engineering practice, Vol. 4, No. 7, 1996, pp. 929-937

[VHJ1999] **Varsamidis, T., Hope, S., Jobling, C.,** *Use of a prototype CACE integration framework based on the unified information model* // IEEE international symposium on computer-aided control system design, August 1999, pp. 392-397

[Vir1994] *Virtual reality* // McGraw-Hill dictionary of scientific and technical terms, 5th ed., S.P. Parker (ed.), McGraw-Hill, 1994, pp. 2136-2137

[Vir1996] *Virtual* // Webster's encyclopedic unabridged dictionary of the English language, New York, Gramercy books, 1996, p. 1592

[Vir2004] *Virtual reality* / S. Bryson // Computer science handbook / A.B. Tucker (ed.), Chapman & Hall – CRC, 2004, p. 42-I

[Vir2005] *Virtual* // Webopedia, 2005, [www.webopedia.com]

[W3C2005] *W3 Consortium & XML (Extensible Markup Language),* 2005, [www.w3.org/XML]

[WB2000] **Wilson, D.I., Balderud, J.,** *Experiences building and validating a large scale dynamic paper machine model* // IEE conference on model validation for plant control and condition monitoring, London, March 2000, pp. 2/1-2/7

[Wen2002] **Wenderoth, A.,** *Nokia's way to a digital manufacturing company* // Delmia European user conference, Stuttgart, 2002, (PDF document)

[Wie1948] **Wiener, N.,** *Cybernetics, or, control and communication in the animal and the machine,* New York, 1948, 194 p.

[WML1998] **Wilson, B.H., Mourant, R.R., Li, M., Xu, W.,** *A virtual environment for training overhead crane operators: Real-time implementation* // IEE transactions, No. 30, 1998, pp. 589-595

[WT1997] **Wilson, H.B., Turcotte, L.H.,** *Advanced mathematics and mechanics applications using MATLAB*, 2^{nd} ed., CRC Press, 1997, 632 p.

[YC1993] **Yam, Y., Chiang, R.Y.,** *Integrated identification/discrete sup* H_∞ *synthesis for spacecraft pointing control* // IEEE American control conference, Vol. 3, 1993, pp. 3048-3051

[Yur2001] **Yurevich, E.I.,** *Robotics (Robototekhnika)*: Textbook, St. Petersburg, 2001, 300 p., (in Russian)

[Zak2003] **Żak, S.H.,** *Systems and control*, Oxford University Press, 2003, 704 p.

[ZN1942] **Ziegler, J.G., Nichols, N.B.,** *Optimum settings for automatic controllers* // Transactions of ASME, 64, 1942, p. 759

[ZY2000] **Zenkevich, S.L., Yuschenko, A.S.,** *Robot control (Upravleniye robotami: Osnovy upravleniya manipulyatsionnymi robotami)*, Moscow state technical university of N. Bauman, 2000, 400 p., (in Russian)

[ZY2003] **Zlygarev, V.A., Yurin, V.N.,** *Performance analysis of the engineering for constructing CALS (Analiz proizvoditelnosty inzhenernogo truda dlya postroeniya CALS)* // The 5^{th} international conference-forum "Applications of CALS for improving the quality and competitiveness of the science-intensive products", 2003, (in Russian)

APPENDIX 1. SELECTING THE VISCOUS FRICTION COEFFICIENTS FOR THE GANTRY CRANE REFERENCE MODEL

Let the damping ratio of the spreader-load subsystem be about 1% of the critical damping[1]. Using the third equation in (5.11) and considering the trolley position and the rope length are constant, we obtain the transfer function from the external torque[2], which is applied to the load, to the load displacement angle. The denominator of the transfer function is $s^2 + \frac{k_3}{ML_0^2}s + \frac{g}{L_0}$. The discriminant is zero when the damping ratio is critical [Bur2001], yielding $k_3^{critical} = 2ML_0^2\sqrt{g/L_0}$. Noting that $k_3 = 0.01 \cdot k_3^{critical}$ and applying the numerical values of the parameters used in our example, we obtain $k_3 \approx 3.9 \cdot 10^5\, {kg \cdot m^2}/{s}$. The value of k_3 corresponds to the oscillatory transient process with the settling time about 450 *s*.

The damping ratio of the trolley requires a different selection process because of the zero free term in the denominator of its transfer function[3]. Let the damped transient process settling time be $T_s = 45\, s$ assuming the effect of the trolley viscous friction is ten times more than one of the spreader-load subsystem. From the first equation in (5.11), the non-zero pole of the trolley transfer function is $-\frac{k_1}{\left(J_1 + (M+m)b_1^2\right)}$. Hence, $k_1 \approx 3\frac{\left(J_1 + (M+m)b_1^2\right)}{T_s}$. Applying the numerical values of the parameters, we obtain $k_1 \approx 0.92\, {kg \cdot m^2}/{s}$. Due to the physical similarity of the viscous friction in the trolley and the hoist drives, we have $k_1 \approx k_2$.

To sum up, $k_1 = 0.92\, {kg \cdot m^2}/{s}$, $k_2 = 0.92\, {kg \cdot m^2}/{s}$, $k_3 = 3.9 \cdot 10^5\, {kg \cdot m^2}/{s}$.

The significant distinction of (1) the angular velocity of the trolley and the hoist drive motors and (2) the angular velocity of the spreader-load subsystem leads to the five-orders-of-magnitude difference between k_1, k_2 and k_3.

It has to be noted that the described selection procedure is rather coarse. The exact values of the damping ratio coefficients do not matter in our example. Important are their orders.

---

[1] Based on the assessments given in [ANM2003]
[2] Equals zero in (5.11)
[3] Physically it means that the trolley is non-asymptotically stable

APPENDIX 2. MATLAB AND IGRIP PROGRAMS FOR SIMULATING THE GANTRY CRANE AND DESIGNING ITS CONTROL SYSTEM

Listing 1. Comparing the nonlinear and the linearized models in one of the GC work regimes[1] (**lin_vs_nonlin.m**).

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% The analytical non-linear model vs. the analytical linear one
%
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

function [] = lin_vs_nonlin( )

% Obtain the static parameters of the crane
[Ts, J1, b1, K1, V1_max, J2, b2, K2, V2_max, K3, L0, m, M, g] = crane_parameters;

% Generate input torques
Tacc = 2;                    % Acceleration time
Tfree = 4;                   % Free sways time
Ttotal = 25;                 % Total simulation time

amp = 1030 * 0.5;            % Trolley drive's torque amplitude

k = 1;                       % Counter

u1 = zeros( Ttotal / Ts, 1 );
u1( 1 : Tacc / Ts ) = amp;
u1( Tacc / Ts + 1 : Tacc / Ts + Tfree / Ts ) = 0;
u1( Tacc / Ts + Tfree / Ts + 1 : ( Tacc * 2 + Tfree ) / Ts ) = -amp;
u1( ( Tacc * 2 + Tfree ) / Ts + 1 : end ) = 0;

Tacc = 2;                    % Acceleration time

Teq = M * g * b2;            % Balancing torque
Tmax = (M * b2 + J2 / b2 ) * V2_max / Tacc * 0.5;  % The maximum relative hoist/lower torque
u2 = zeros( length( u1 ), 1 );

u2( 1 : Tacc / Ts ) = - Tmax - Teq;
```

[1] The extension of the MATLAB files is ".m", the extension of the IGRIP files is ".gsl"

```
u2( Tacc / Ts : end ) = - Teq;

% Simulate the NON-LINEAR model
[T_nl,Y_nl] = nonlinear_model_sim( [u1 u2] );

% Simulate the LINEAR model
sys = lin_model;                    % Obtain the analytical linear model

u2 = u2 + Teq;                      % NOTE: this is a linear model -> the balancing torque has to be removed

[Y, T, X] = lsim( sys, [u1 u2],  [ 0 : Ts : ( length( [u1 u2] ) - 1 ) * Ts ] ); % Simulation

% Compare the results
figure;
subplot(3,1,1)
plot( T_nl, ( Y_nl(:,1) - 694.44 ) * b1, 'r-.', T, Y(:,1) * b1, 'b' ); grid;
xlabel( 'Time, s' );
ylabel( 'Trolley relative position, m' );
legend( 'Nonlinear model', 'Linearized model' );

subplot(3,1,2)
plot( T_nl, ( Y_nl(:,2) - 2295.1 ) * b2, 'r-.', T, Y(:,2) * b2, 'b' ); grid;
xlabel( 'Time, s' );
ylabel( 'Load relative height, m' );

subplot(3,1,3)
plot( T_nl, Y_nl(:,3) * 180 / 3.1415, 'r-.', T, Y(:,3) * 180 / 3.1415, 'b' ); grid;
xlabel( 'Time, s' );
ylabel( 'Load angular displacement, deg' );

% Angular displacement of the load in the phase plane
figure;
plot( Y_nl(2:end,3) * 180 / 3.1415, diff( Y_nl(:,3) ) / Ts * 180 / 3.1415, 'r-.',...
   Y(2:end,3) * 180 / 3.1415, diff( Y(:,3) ) / Ts * 180 / 3.1415, 'b' ); grid;
title( 'Load angular displacement' )
xlabel( 'Position, deg' );
ylabel( 'Velocity, deg/s' );
legend( 'Nonlinear model', 'Linearized model' );
```

```
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% The static parameters of the crane
%
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

function [Ts, J1, b1, K1, V1_max, J2, b2, K2, V2_max, K3, L0, m, M, g] = crane_parameters()

Ts = 0.1;                  % Sampling period

% Trolley drive
J1 = 3.75066;              % Moment of inertia
b1 = 1.44e-2;              % Equivalent radius of the drum reduced to the motor side
K1 = 0.92;                 % Viscous friction coefficient
V1_max = 2.5;              % Maximum trolley velocity

% Hoist drive
J2 = 78.5;                 % Moment of inertia
b2 = 1.22e-2;              % Equivalent radius of the drum reduced to the motor side
K2 = 0.92;                 % Viscous friction coefficient
V2_max = 1.0;              % Maximum hoist/lower velocity

% Spreader-load subsystem
K3 = 4.0e5;                % Viscous friction coefficient
L0 = 28;                   % Nominal length of the rope

m = 6000;                  % Mass of the trolley
M = 42500;                 % Mass of the load
g = 9.81;                  % Gravity acceleration


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% The analytical linear model of the crane
%
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

function [sys] = lin_model( )

% Obtain the static parameters of the crane
[Ts, J1, b1, K1, V1_max, J2, b2, K2, V2_max, K3, L0, m, M, g] = crane_parameters;

% The model in the state space
```

```
a43 = M * L0 * g * b1 / ( L0 * ( J1 + b1^2 * m ) );
a44 = - K1 / ( J1 + b1^2 * m );
a46 = K3 * b1 / ( L0 * ( J1 + b1^2 * m ) );
a55 = - K2 / ( J2 + M * b2^2 );
a63 = - g * ( J1 + b1^2 * ( m + M ) ) / ( L0 * ( J1 + b1^2 *m ) );
a64 = K1 * b1 / ( L0 * ( J1 + b1^2 * m ) );
a66 = - K3 * ( J1 + b1^2 * ( m + M ) ) / ( M * L0^2 * ( J1 + b1^2 * m ) );

A = [ 0 0 0 1 0 0; 0 0 0 0 1 0; 0 0 0 0 0 1; ...
    0 0 a43 a44 0 a46; 0 0 0 0 a55 0; 0 0 a63 a64 0 a66 ];

B = [ 0 0; 0 0; 0 0; 1 / ( J1 + b1^2 * m ) 0; 0 1 / ( J2 + b2^2 * M ); - b1 / ( L0 * ( J1 + b1^2 * m ) ) 0 ];
C = [ 1 0 0 0 0 0; 0 1 0 0 0 0; 0 0 1 0 0 0 ];
D = 0;

sys_c = ss( A, B, C, D );

% Descretize the system
sys = c2d( sys_c, Ts, 'zoh');



%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Simulation of the non-linear model of the crane
% This function uses:
%  - nolinear_model_cauchy: 'ode45' solves non-linear Lagrange's equations
%
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

function [T_nl,Y_nl] = nonlinear_model_sim(u)

```
global input_torque;

Ts = crane_parameters;                    % Sampling period

input_torque = u;

% Simulation for identification
```

[T_nl,Y_nl] = ode45(**@nonlinear_model_cauchy**, [0 : Ts : (length(input_torque) - 1) * Ts], [694.44 2295.1 0 0 0 0], odeset('RelTol', 1e-6));

```
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Simulation of the non-linear model of the crane
% This function is used in ode45
%
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

function [dx] = nonlinear_model_cauchy( t, x )

global input_torque;

% Obtain the static parameters of the crane
[Ts, J1, b1, K1, V1_max, J2, b2, K2, V2_max, K3, L0, m, M, g] = crane_parameters;

index = floor( t / Ts ) + 1;
T1 = input_torque(index,1);
T2 = input_torque(index,2);

% Update state variables
x1 = x(1);
x2 = x(2);
x3 = x(3);
x1_1 = x(4);
x2_1 = x(5);
x3_1 = x(6);

% Compute x1_2, x2_2, x3_2
A11 = J1 + ( M + m ) * b1^2;
A12 = M * b1 * b2 * sin( x3 );
A13 = M * b1 * b2 * x2 * cos( x3 );
B1 = M * b1 * b2 * ( - 2 * x2_1 * x3_1 * cos( x3 ) + x2 * x3_1^2 * sin( x3 ) ) - K1 * x1_1 + T1;

A21 = M * b1 * b2 * sin( x3 );
A22 = J2 + M * b2^2;
A23 = 0;
B2 =  M * b2^2 * x2 * x3_1^2 + M * g * b2 * cos( x3 ) - K2 * x2_1 + T2;

A31 = M * b1 * b2 * x2 * cos( x3 );
A32 = 0;
A33 = M * b2^2 * x2^2;
B3 = - 2 * M * b2^2 * x2 * x2_1 * x3_1 - M * g * b2 * x2 * sin( x3 ) - K3 * x3_1;
```

```
Q = [A11 A12 A13; A21 A22 A23; A31 A32 A33] \ [B1;B2;B3];

% Compute an explicit Cauchy's form
dx = [ x1_1; x2_1; x3_1; Q(1); Q(2); Q(3) ];
```

Listing 2. The waveforms and the spectra of the input signals used in the identification experiment (**input_gen.m**).

```
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Generate the input for the identification experiment
%
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

function [] = input_gen()

% Obtain the static parameters of the crane
[Ts, J1, b1, K1, V1_max, J2, b2, K2, V2_max, K3, L0, m, M, g] = crane_parameters;

%~~~~~~~~~~~~~~~~~~~~~~~~~
% DATA FOR IDENTIFICATION
%~~~~~~~~~~~~~~~~~~~~~~~~~

Ttotal = 81;                % Total length of the simulation

% Generate T1: square wave
%~~~~~~~~~~~~~~~~~~~~~~~~~~
Tacc = 1.0;                 % Duration of: acceleration
Tconst = 3.0;               % Constant velocity
Tfree = 8.0;                % Free sways

amp = 1030 * 0.5;           % Trolley drive's torque amplitude

n = 0.1 * amp;              % Amplitude of noise

counter = 1;                % Counter

cycles = 3;                 % Number of cycles

u1 = zeros( 1, Ttotal / Ts );
```

```
% Pattern (duration, torque)
pattern = [ Tacc, amp;...              % Accelerate to the right
      Tconst, 0;...                    % Constant speed
      Tacc, -amp;...                   % Decelerate
      Tfree, 0;...                     % Constant speed (zero)
      Tacc, -amp;...                   % Accelerate to the left
      Tconst, 0;...                    % Constant speed
      Tacc, amp;...                    % Decelerate
      Tfree, 0 ];                      % Constant speed (zero)

for C = 1:cycles
   for J = 1:size(pattern,1)
      for I = 1:pattern(J,1)/Ts
         u1( counter ) = pattern(J,2);
         counter = counter + 1;
      end
   end
end

% Add noise
randn( 'state', sum( 100 * clock ) );              % Reset the state
u1 = u1 + randn( size( u1 ) ) * n;

% Plot the input signal
figure; subplot( 2, 2, 1 );
plot( [ 0 : Ts : ( length( u1 ) - 1 ) * Ts ], u1 ); grid;
title( 'Signal T1' );
xlabel( 'Time, s' );
ylabel( 'Amplitude, Nm' );

% Compute Fourier transformation & plot it
N = 2 ^ floor( log2( length( u1 ) ) );             % Number of points (power of 2)
t = [ 0 : Ts :( N - 1 ) * Ts ];                    % Time vector
pow_spectrum = abs( fft( u1, N ) / N ).^2;         % Power spectrum. To obtain the amplitude of the
                                                   % harmonics: sqrt(pow_spectrum)*2. "2" is due to the fact
                                                   % that the energy in “fft” is distributed between frequencies
                                                   % "-f" and "f"
subplot( 2, 2, 2 );
stem( [ 0 : N / 2 - 1 ] / N / Ts , pow_spectrum( 1 : N / 2 ) ); grid;
title( 'Signal T1. Power spectrum' );
xlabel( 'Frequency, Hz' );
```

```
ylabel( 'Power, (Nm)^2' );

% Remove the temporary vector variables
clear t pow_spectrum;

% Generate T2: square wave
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Tacc = 2.0;                                          % Duration of acceleration

Teq = M * g * b2;                                    % Balancing torque
Tmax = (M * b2 + J2 / b2 ) * V2_max / Tacc * 0.8;    % The maximum relative torque
H_amp = - Teq - Tmax;                                % Hoist torque amplitude
L_amp = - Teq + Tmax;                                % Lower torque amplitude

n = 0.01 * Tmax;                                     % Amplitude of noise

counter = 1;                                         % Counter

cycles = 4;

u2 = zeros( 1, Ttotal / Ts ) - Teq;

% Pattern (duration, torque)
pattern = [ Tacc, -Teq;...                           % Delay
        Tacc, H_amp;...                              % Accelerate upwards
        Tacc, L_amp;...                              % Decelerate
        Tacc, L_amp;...                              % Accelerate downwards
        Tacc, H_amp;...                              % Decelerate
        Tacc, -Teq;...                               % Delay
        Tacc, L_amp;...                              % Accelerate downwards
        Tacc, H_amp;...                              % Decelerate
        Tacc, H_amp;...                              % Accelerate upwards
        Tacc, L_amp];                                % Decelerate

for C = 1:cycles
  for J = 1:size(pattern,1)
    for I = 1:pattern(J,1)/Ts
      u2( counter ) = pattern(J,2);
      counter = counter + 1;
    end
  end
```

```
end

% Add noise
randn( 'state', sum( 100 * clock ) );            % Reset the state
u2 = u2 + randn( size( u2 ) ) * n;

% Plot the input signal
subplot( 2, 2, 3 );
plot( [ 0 : Ts : ( length( u2 ) - 1 ) * Ts ], u2 ); grid;
title( 'Signal T2' );
xlabel( 'Time, s' );
ylabel( 'Amplitude, Nm' );

% Compute Fourier transformation & plot it
N = 2 ^ floor( log2( length( u2 ) ) );           % Number of points (power of 2)
t = [ 0 : Ts :( N - 1 ) * Ts ];                  % Time vector
pow_spectrum = abs( fft( u2 + Teq, N ) / N ).^2; % Power spectrum (without constant component Teq)

subplot( 2, 2, 4 );
stem( [ 0 : N / 2 - 1 ] / N / Ts , pow_spectrum( 1 : N / 2 ) ); grid;
title( 'Signal T2. Power spectrum' );
xlabel( 'Frequency, Hz' );
ylabel( 'Power, (Nm)^2' );

% Check the persistency of excitation
L = length( u1 );                                % Total number of measurements
Imax = 10;                                       % Size of matrix U (number of rows, Imax > model order )
m = 2;                                           % Dimension of the input

Jmax = L - 2*Imax + 1;                           % Number of columns
U = zeros( 2*m*Imax, Jmax );

for I = 1 : 2*Imax
   for J = 1 : Jmax
      U( m*I-1, J ) = u1( I+J-1 );
      U( m*I, J ) = u2( I+J-1 );
   end
end

% For PE of order "2*Imax" the rank of U must be 2*m*Imax (full rank)
r = rank( U * U' )
```

```
Imax

% Save to a text file
fid = fopen( 'input_data_id.txt', 'wt');
fprintf( fid, '%1.4f %1.4f\n', [u1; u2] );
fclose(fid);

% Remove the temporary vector variables
clear t pow_spectrum u1 u2;

%~~~~~~~~~~~~~~~~~~~~
% DATA FOR VALIDATION
%~~~~~~~~~~~~~~~~~~~~

Ttotal = 85;                    % Total length of the simulation

% Generate T1: square wave
%~~~~~~~~~~~~~~~~~~~~~~~~~
Tacc = 1.5;                     % Duration of acceleration
Tconst = 4.0;                   % Constant velocity
Tfree = 6.5;                    % Free sways

amp = 1030 * 0.6;               % Trolley drive's torque amplitude

n = 0.1 * amp;                  % Amplitude of noise

counter = 1;                    % Counter

cycles = 3;                     % Number of cycles

u1 = zeros( 1, Ttotal / Ts );

% Pattern (duration, torque)
pattern = [ Tacc, amp;...       % Accelerate to the right
    Tconst, 0;...               % Constant speed
    Tacc, -amp;...              % Decelerate
    Tfree, 0;...                % Constant speed (zero)
    Tacc, -amp;...              % Accelerate to the left
    Tconst, 0;...               % Constant speed
    Tacc, amp;...               % Decelerate
    Tfree, 0 ];                 % Constant speed (zero)
```

```
for C = 1:cycles
   for J = 1:size(pattern,1)
      for I = 1:pattern(J,1)/Ts
         u1( counter ) = pattern(J,2);
         counter = counter + 1;
      end
   end
end

% Add noise
randn( 'state', sum( 100 * clock ) );              % Reset the state
u1 = u1 + randn( size( u1 ) ) * n;

% Plot the input signal
figure; subplot( 2, 2, 1 );
plot( [ 0 : Ts : ( length( u1 ) - 1 ) * Ts ], u1 ); grid;
title( 'Signal T1' );
xlabel( 'Time, s' );
ylabel( 'Amplitude, Nm' );

% Compute Fourier transform & plot it
N = 2 ^ floor( log2( length( u1 ) ) );              % Number of points (power of 2)
t = [ 0 : Ts :( N - 1 ) * Ts ];                     % Time vector
pow_spectrum = abs( fft( u1, N ) / N ).^2;          % Power spectrum

subplot( 2, 2, 2 );
stem( [ 0 : N / 2 - 1 ] / N / Ts , pow_spectrum( 1 : N / 2 ) ); grid;
title( 'Signal T1. Power spectrum' );
xlabel( 'Frequency, Hz' );
ylabel( 'Power, (Nm)^2' );

% Remove from memory temporary vector variables
clear t pow_spectrum;

% Generate T2: square wave
%~~~~~~~~~~~~~~~~~~~~~~~~~~
Tacc = 2.0;                                         % Duration of: acceleration

Teq = M * g * b2;                                   % Balancing torque
Tmax = (M * b2 + J2 / b2 ) * V2_max / Tacc * 0.7;   % The maximum relative torque
```

```
H_amp = - Teq - Tmax;                    % Hoist torque amplitude
L_amp = - Teq + Tmax;                    % Lower torque amplitude

n = 0.01 * Tmax;                         % Amplitude of noise

counter = 1;                             % Counter

cycles = 4;

u2 = zeros( 1, Ttotal / Ts ) - Teq;

% Pattern (duration, torque)
pattern = [ Tacc, -Teq;...               % Delay (wait until the trolley stops accelerating)
            Tacc, H_amp;...              % Accelerate upwards
            Tacc, L_amp;...              % Decelerate
            Tacc, L_amp;...              % Accelerate downwards
            Tacc, H_amp;...              % Decelerate
            Tacc, -Teq;...               % Delay
            Tacc, L_amp;...              % Accelerate downwards
            Tacc, H_amp;...              % Decelerate
            Tacc, H_amp;...              % Accelerate upwards
            Tacc, L_amp];                % Decelerate

for C = 1:cycles
   for J = 1:size(pattern,1)
      for I = 1:pattern(J,1)/Ts
         u2( counter ) = pattern(J,2);
         counter = counter + 1;
      end
   end
end

% Add noise
randn( 'state', sum( 100 * clock ) );    % Reset the state
u2 = u2 + randn( size( u2 ) ) * n;

% Plot the input signal
subplot( 2, 2, 3 );
plot( [ 0 : Ts : ( length( u2 ) - 1 ) * Ts ], u2 ); grid;
title( 'Signal T2' );
xlabel( 'Time, s' );
```

```
ylabel( 'Amplitude, Nm' );

% Compute Fourier transformation & plot it
N = 2 ^ floor( log2( length( u2 ) ) );              % Number of points (power of 2)
t = [ 0 : Ts :( N - 1 ) * Ts ];                     % Time vector
pow_spectrum = abs( fft( u2 + Teq, N ) / N ).^2;    % Power spectrum (without constant component Teq)

subplot( 2, 2, 4 );
stem( [ 0 : N / 2 - 1 ] / N / Ts , pow_spectrum( 1 : N / 2 ) ); grid;
title( 'Signal T2. Power spectrum' );
xlabel( 'Frequency, Hz' );
ylabel( 'Power, (Nm)^2' );

% Save to a text file
fid = fopen( 'input_data_val.txt', 'wt');
fprintf( fid, '%1.4f %1.4f\n', [u1; u2] );
fclose(fid);
```

Listing 3. Generating the data-based dynamic model of the GC by simulating the GC DVP in IGRIP. The input signal is produced by MATLAB function input_gen.m (**gantry_sim.gsl**).

```
-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
PROGRAM gantry_sim
-- -- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

VAR
  q_initial : ARRAY[3] of REAL
  step_counter : REAL
  amp : REAL
  sampling_period : REAL
  torque1 : REAL
  torque2 : REAL
  data_type : INTEGER

BEGIN MAIN

  $Stepsize = 0.1
  $Motype = JOINT
  $DYN_STATUS = DYN_OFF
```

```
  sampling_period = 0.1
  step_counter = 1

  -- Initialize the position of the device (x1 = 10m; x2 = 28m. NB: x2min == 14m).
  q_initial[0] = 39788.74
  q_initial[1] = 131498.51
  q_initial[2] = 0

  -- Move to the initial position at the origin
  MOVE JOINTS TO q_initial
  SIM_UPDATE

  -- Prepare for the simulation
  $DYN_STATUS = DYN_SIMULATION
  $DYN_TERM_CHECK = FALSE

  -- File that contains persistently exciting input
  OPEN FILE 'c:/igrip/input_data.txt' FOR TEXT INPUT AS 1

  -- File to store the simulation data
  OPEN FILE 'c:/igrip/output_data.txt' FOR TEXT OUTPUT AS 2

  -- Read the torque data until EOF & apply the torque
  WHILE ( ( data_type = read( #1, torque1, torque2 ) )  <> EOF ) DO

    -- Write joints' positions
    write(  #2,  (step_counter-1)*sampling_period,  ' ',  jointval(1)-39788.74,  ' ',  jointval(2)-131498.51,  ' ',
jointval(3), CR )

    -- Apply the torque
    part_ext_moment( 'kobo_crane:drumT', vec(0,torque1,0) )
    part_ext_moment( 'kobo_crane:drumH', vec(0,torque2,0) )
    delay sampling_period * 1000
    step_counter = step_counter + 1

  ENDWHILE

  CLOSE #1
  CLOSE #2

END MAIN
```

Listing 4. Identification of the GC linear model using the data-based dynamic model generated in IGRIP (**crane_id.m**).

```
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Identification of the crane model
%
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

function [sys_id] = crane_id( )

% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Identification
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Obtain the static parameters of the crane
[Ts, J1, b1, K1, V1_max, J2, b2, K2, V2_max, K3, L0, m, M, g] = crane_parameters;

% Read the input torque data from the text file
fid = fopen('input_data_id.txt', 'rt' );
u = (fscanf( fid, '%f %f', [2,inf] ))';
fclose( fid );

u(:,2) = u(:,2) + M * g * b2;                  % NOTE: for the linear model we have to remove
                                               % the balancing torque
% Obtain simulation data from IGRIP. Convert to radians
y = sim_data_crane;
y(:,2) = y(:,2) * 3.1415926 / 180;
y(:,3) = y(:,3) * 3.1415926 / 180;
y(:,4) = y(:,4) * 3.1415926 / 180;

Y = [y(:,2) y(:,3) y(:,4)];

% Compute derivatives (Lagrange's polynomial of the second order)
for J = 1:3
   N = size( Y(:,J), 1 );
   Y1(1,J) = ( -3 * Y(1,J) + 4 * Y(2,J) - Y(3,J) ) / ( 2 * Ts );          % First node
   for I = 2:N-1
      Y1(I,J) = ( Y(I+1,J) - Y(I-1,J) ) / ( 2 * Ts  );
   end
   Y1(N,J) = ( Y(N-2,J) - 4 * Y(N-1,J) + 3 * Y(N,J) ) / ( 2 * Ts );       % Last node
end
```

```
% Estimate the state-space models using the subspace method
data = iddata( [Y Y1], u, Ts );                                    % Generalized input
m = n4sid( data, [2:1:10] ,'DisturbanceModel', 'None', 'N4Horizon', [30 60 60] );

% Obtain the state-space model
[A, B, C, D, K, X0] = ssdata(m);
sys_id = ss( A, B, C, D, Ts );

eig( sys_id )                          % Eigenvalues of the identified linear model
eig( lin_model )                       % Eigenvalues of the analytical linear model

% Simulate the identified model
t = [ 0 : Ts : ( length( u ) - 1 ) * Ts ];
[YS,TS,XS] = lsim( sys_id, u, t );

% Plot the results
figure;

subplot( 3, 1, 1 ); plot( TS, YS(:,1)* b1, y(:,1), y(:,2) * b1, 'r' ); grid;
title( 'Identification results: Joint 1' );
xlabel( 'Time, s' ); ylabel( 'Position, m' );
legend( 'Identified model', 'Identification data' );

subplot( 3, 1, 2 ); plot( TS, YS(:,2) * b2, y(:,1), y(:,3) * b2, 'r' ); grid;
title( 'Identification results: Joint 2' );
xlabel( 'Time, s' ); ylabel( 'Position, m' );

subplot( 3, 1, 3 ); plot( TS, YS(:,3)*180/3.1415926, y(:,1), y(:,4)*180/3.1415926, 'r' ); grid;
title( 'Identification results: Joint 3' );
xlabel( 'Time, s' ); ylabel( 'Position, deg' );

% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Validation
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Read the input torque data from the text file
fid = fopen('input_data_val.txt', 'rt' );
u_val = (fscanf( fid, '%f %f', [2,inf] ))';
fclose( fid );
```

```
u_val(:,2) = u_val(:,2) + M * g * b2;                % NOTE: for the linear model we have to
                                                      % remove the balancing torque

% Obtain the simulation data from IGRIP. Convert to radians
y_val = val_data_crane;
y_val(:,2) = y_val(:,2) * 3.1415926 / 180;
y_val(:,3) = y_val(:,3) * 3.1415926 / 180;
y_val(:,4) = y_val(:,4) * 3.1415926 / 180;

% Simulate the identified linear model on the validation data set
t_val = [ 0 : Ts : ( length( u_val ) - 1 ) * Ts ];
[YS_val,TS_val,XS_val] = lsim( sys_id, u_val, t_val );

% Simulate the analytical linear model on the validation data set
t_val = [ 0 : Ts : ( length( u_val ) - 1 ) * Ts ];
[YS_val_a,TS_val_a,XS_val_a] = lsim( lin_model, u_val, t_val );

% Plot the results
figure;

% IGRIP validation data vs. identified linear model
subplot( 3, 2, 1 ); plot( TS_val, YS_val(:,1)*b1, y_val(:,1), y_val(:,2)*b1, 'r:' ); grid;
title( 'Validation results: Joint 1' );
xlabel( 'Time, s' ); ylabel( 'Position, m' );
legend( 'Identified model', 'Validation data' );

subplot( 3, 2, 3 ); plot( TS_val, YS_val(:,2)*b2, y_val(:,1), y_val(:,3)*b2, 'r:' ); grid;
title( 'Validation results: Joint 2' );
xlabel( 'Time, s' ); ylabel( 'Position, m' );

subplot( 3, 2, 5 ); plot( TS_val, YS_val(:,3)*180/3.1415926, y_val(:,1), y_val(:,4)*180/3.1415926, 'r:' ); grid;
title( 'Validation results: Joint 3' );
xlabel( 'Time, s' ); ylabel( 'Position, deg' );

% Identified linear model vs. analytical linear model
subplot( 3, 2, 2 ); plot( TS_val, YS_val(:,1)*b1, TS_val_a, YS_val_a(:,1)*b1, 'r:' ); grid;
title( 'Identified linear model vs. analytical linear model: Joint 1' );
xlabel( 'Time, s' ); ylabel( 'Position, m' );
legend( 'Identified model', 'Analytical model' );

subplot( 3, 2, 4 ); plot( TS_val, YS_val(:,2)*b2, TS_val_a, YS_val_a(:,2)*b2, 'r:' ); grid;
title( 'Identified linear model vs. analytical linear model: Joint 2' );
```

```
xlabel( 'Time, s' ); ylabel( 'Position, m' );

subplot( 3, 2, 6 ); plot( TS_val, YS_val(:,3)*180/3.1415926, TS_val_a, YS_val_a(:,3)*180/3.1415926, 'r:' );
grid;
title( 'Identified linear model vs. analytical linear model: Joint 3' );
xlabel( 'Time, s' ); ylabel( 'Position, deg' );

% Optional: compute MDL (AIC) to estimate the order of the model
```

Listing 5. Simulating the controlled GC in IGRIP (**crane_control.gsl**).

```
-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
PROGRAM crane_control
-- LQ control
-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

CONST

  NUM_DOF 3                                   -- Number of degrees-of-freedom
  NUM_IN 2                                    -- Number of inputs
  NUM_STATE 6                                 -- Number of states
  sampling_period 0.1                         -- Sampling period
  total_steps 200                             -- Total number of steps

VAR
  q_initial : ARRAY[NUM_DOF] of REAL          -- Nominal position of the device
  K : ARRAY[NUM_IN,NUM_STATE] of REAL         -- Feedback matrix
  torque_amp : ARRAY[NUM_IN] of REAL          -- Applied torque
  X : ARRAY[NUM_STATE] of REAL                -- State
  X_p : ARRAY[NUM_DOF] of REAL                -- The coordinates at the previous step
  Ref : ARRAY[NUM_STATE] of REAL              -- Reference state
  step_counter : REAL                         -- Step counter
  Tbal : REAL                                 -- Balancing torque

BEGIN MAIN

  $Stepsize = sampling_period
  $Motype = JOINT
  $DYN_STATUS = DYN_OFF
  step_counter = 1
```

```
Tbal = 42500 * 9.81 * 1.22e-2

-- Initialize the nominal position of the device (in degrees; x1 = 10m; x2 = 28m. NB: x2min == 14m).
q_initial[0] = 39788.74
q_initial[1] = 131498.51
q_initial[2] = 0

-- Initialize the feedback matrix
K[0,0] = 3.00977
K[0,1] = -0.0527380
K[0,2] = 1741.48
K[0,3] = 7.93292
K[0,4] = 1.58774
K[0,5] = 6212.08
K[1,0] = 0.241803
K[1,1] = 9.53668
K[1,2] = -53.6705
K[1,3] = 0.0809296
K[1,4] = 28.7878
K[1,5] = 165.761

-- Initialize the reference state (relative, in radians; x1 = 2m (move right), x2 = -2m (hoist) )
Ref[0] = 2 / 1.44e-2
Ref[1] = -2 / 1.22e-2
Ref[2] = 0
Ref[3] = 0
Ref[4] = 0
Ref[5] = 0

-- Initialize the previous-step state vector (coordinates only)
X_p[0] = 0
X_p[1] = 0
X_p[2] = 0

-- Move to the initial position at the origin
MOVE JOINTS TO q_initial
SIM_UPDATE

-- Prepare for the simulation
$DYN_STATUS = DYN_SIMULATION
$DYN_TERM_CHECK = FALSE
```

```
  -- File to store the simulation data
  OPEN FILE 'c:/igrip/output_data_control.txt' FOR TEXT OUTPUT AS 1

  FOR step_counter = 1 TO total_steps DO
    -- Build the state vector

    -- The coordinates ( convert deg -> rad )
    X[0] = ( jointval(1) - q_initial[0] ) * 3.1415927 / 180
    X[1] = ( jointval(2) - q_initial[1] ) * 3.1415927 / 180
    X[2] = ( jointval(3) - q_initial[2] ) * 3.1415927 / 180

    -- and their derivatives
    X[3] = ( X[0] - X_p[0] ) / sampling_period
    X[4] = ( X[1] - X_p[1] ) / sampling_period
    X[5] = ( X[2] - X_p[2] ) / sampling_period

    -- Update the previous-step coordinates
    X_p[0] = X[0]
    X_p[1] = X[1]
    X_p[2] = X[2]

    -- Compute the control signal
    torque_amp[0] = K[0,0]*(Ref[0]-X[0]) + K[0,1]*(Ref[1]-X[1]) + K[0,2]*(Ref[2]-X[2]) + K[0,3]*(Ref[3]-
X[3]) + K[0,4]*(Ref[4]-X[4]) + K[0,5]*(Ref[5]-X[5])
    torque_amp[1] = K[1,0]*(Ref[0]-X[0]) + K[1,1]*(Ref[1]-X[1]) + K[1,2]*(Ref[2]-X[2]) + K[1,3]*(Ref[3]-
X[3]) + K[1,4]*(Ref[4]-X[4]) + K[1,5]*(Ref[5]-X[5]) - Tbal

    -- Write joints' positions etc.
    write( #1, step_counter * sampling_period, ' ', X[0], ' ',  X[1], ' ', X[2], ' ', torque_amp[0], ' ', torque_amp[1],
CR )

    -- Apply the torque
    part_ext_moment( 'kobo_crane:drumT', vec( 0, torque_amp[0], 0 ) )
    part_ext_moment( 'kobo_crane:drumH', vec( 0, torque_amp[1], 0 ) )
    delay sampling_period*1000

  ENDFOR

  CLOSE #1
END MAIN
```

Helsinki University of Technology

Industrial Information Technology Laboratory Publications

ISBN: 951-22-8167-8 (printed version)

ISBN: 951-22-8168-6 (electronic version)

ISSN: 1459-6458